USER-ENTITY DIFFERENTIAL PRIVACY IN LEARNING NATURAL LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we introduce a novel concept of user-entity differential privacy (UeDP) to provide formal privacy protection simultaneously to both sensitive entities in textual data and data owners in learning natural language models. To preserve UeDP, we developed a novel algorithm, called UeDP-Alg, optimizing the trade-off between privacy loss and model utility with a tight sensitivity bound derived from seamlessly combining sensitive and non-sensitive textual data together. An extensive theoretical analysis and evaluation show that our UeDP-Alg outperforms baseline approaches in terms of model utility under the same privacy budget consumption on several NLM tasks, using benchmark datasets.

1 INTRODUCTION

Despite remarkable performance in many applications, natural language models (NLMs), such as recurrent neural networks, and LSTMs, are vulnerable to privacy attacks because of such attacks' capacity to memorize unique patterns in training data (Carlini et al., 2018). Recent data training extraction attacks (Carlini et al., 2020) illustrate that sensitive entities, such as an individual person's name, email address, phone number, fax number, physical address, etc., can be accurately extracted from NLM parameters. These sensitive entities and the language data memorized in NLMs may identify a data owner - explicitly by name or implicitly, e.g., via a rare or unique phrase - and link that data owner to extracted sensitive information.

Our main goal is to provide a rigorous guarantee that a trained NLM protects the privacy of data owners' data, namely privacy protection for sensitive entities and the participation information of the data owners in learning NLMs, while maintaining high model utility. The naive solution of anonymizing (including removing/de-identifying) sensitive entities is insufficient and ineffective; since the anonymized (or removed) entities can be matched with non-anonymized data records in a different dataset (Dwork et al., 2014) and the model utility can be significantly affected as shown in our experimental study. While cryptographic approaches can be applied to protect privacy, they introduce a huge computation and resource overhead (Al Badawi et al., 2020). Thus, we proposed to apply differential privacy (Dwork et al., 2006), one of the most adequate solutions, given its formal privacy protection without undue sacrifice in computation efficiency and model utility.

Differential privacy (DP) provides rigorous privacy protection as a probabilistic term, limiting the knowledge about a data record a ML model can leak while learning features of the whole training set. DP-preserving mechanisms have been investigated and applied in practice (Abadi et al., 2016; Phan et al., 2016; Lee & Kifer, 2018; Shokri & Shmatikov, 2015; Yu et al., 2019; Mironov, 2017), including image processing (Phan et al., 2019; Sun & Lyu, 2020; Fan, 2018), healthcare data (Zia et al., 2020; Alnemari et al., 2017; Kartal et al., 2019), financial records (Wu et al., 2019), social media (Wang & Sinnott, 2017; Li et al., 2012; Ou et al., 2018), and NLMs (McMahan et al., 2017; Bagdasaryan et al., 2019; Lyu et al., 2020c;a).

However, existing DP levels of protection, including sample-level DP (Abadi et al., 2016; Roth, 2012; Dwork et al., 2014; Wu et al., 2017; Bassily et al., 2014), user-level DP (McMahan et al., 2017; Ramaswamy et al., 2020), element-level DP (Asi et al., 2019), and local feature-level DP (Lyu et al., 2020a;b; Erlingsson et al., 2014; Duchi et al., 2013), do not provide the privacy protection level demanded to solve our problem. Given training data: 1) Sample-level DP protects privacy of a single sample. 2) User-level DP protects privacy of a single data owner, also called a single user, who may contribute one or more data samples. 3) Element-level DP partitions data owners'

contribution to the training data into sensitive elements, e.g., a curse word, which will be protected. Element-level DP does not provide privacy protection to data owners. And 4) Local (feature-level) DP protects true values of a data sample from being inferred. Recently, Lyu et al. (2020a;b) proposed local DP-preserving approaches for text embedding extraction under (word-level) local DP (**Eq. 2**). However, our theoretical revisiting, confirmed to be correct by the authors Lyu et al., shown that their approaches do NOT achieve word-level DP for extracted text embedding (**Appendix G**) and consume excessive privacy budgets (**Appendix H**). There is a demand for a new level of DP to protect privacy simultaneously for both sensitive entities in the training data and the participation information of data owners in learning NLMs.

Our paper is structured around the following key contributions:

1) We apply DP to NLM training using a new notion of user-entity adjacent datasets (**Definition 2**), leading to formal guarantees of user-entity privacy, rather than privacy for single user or a single sensitive entity.

2) To preserve UeDP, we introduce a novel algorithm, called **UeDP-Alg**, which leverages the recipe of DP-FEDAVG (McMahan et al., 2017) to achieve user-entity adjacent DP via use of the moments accountant (Abadi et al., 2016). Moments accountant was first developed to preserve DP in stochastic gradient descent (SGD) for sample-level privacy. Our federated averaging approach groups multiple SGD updates, which are computed from a two-level random sampling process including a random sample of users and a random sample of sensitive entities, together, thus enabling large-step model updates (**Eq. 4 and Lemma 1**).

3) To address the trade-off between privacy loss and the model utility, we derive a new tight noise scale bound by considering non-sensitive data in learning NLMs without affecting UeDP guarantees (**Lemma 2**). The more non-sensitive data we use to train our model, the less privacy loss for sensitive entities and the higher utility for our model.

4) Through theoretical analysis and rigorous experiments conducted on benchmark datasets, we show that our UeDP-Alg outperforms baseline approaches in terms of model utility on fundamental tasks, i.e., next word prediction and text classification, under the same privacy budget consumption.

2 BACKGROUND

In this section, we revisit NLM tasks, privacy risk in NLMs, and DP. For the sake of clarity, let us focus on next word prediction, and we will extend it to text classification in section 5.

Next Word Prediction. Let D be training data containing U users (data owners), and each user $u \in U$ consists of n_u sentences. Given a vocabulary $\mathcal{V}$, each sentence is a sequence of words, presented as $x = x_1 x_2 \dots x_{m_u}$, where $x_i \in \mathcal{V}, (i \in [1, m_u])$ is a word in x and m_u is the length of x. In next word prediction, the first j words in x, i.e., $x_1, x_2, \dots, x_j$ ($\forall j < m_u$), are used to predict the next word x_{j+1}. Here, x_{j+1} can be considered as a label in the next word prediction task. Perplexity $PP = 2^{-\sum_{x \in D} p(x) \log_2 p(x)}$ is a measurement of how well a model predicts a sample and is often used to evaluate language models, where $p(x)$ is a probability to predict the next word x_{j+1} in x (Chen et al., 1998; Mikolov et al., 2011a). Perplexity is considered the exponential of the cross-entropy loss of a language model; therefore, a lower perplexity indicates a better model. All the notations are summarized in Table 2, **Appendix A**.

Entities and Sensitive Sentences. Each sensitive entity e consists of a word or consecutive words that need to be protected. A list of categories of sensitive entities is in Table 3 (**Appendix B**). For instance, PII related to an identifiable person, such as person names, locations, organizations, phone numbers, can be considered sensitive entities. We denote a user-predefined set of sensitive entities as E. If a sentence x consists of a sensitive entity $e \in E$, x is considered as a sensitive sentence; otherwise, x is a non-sensitive sentence.
For instance, in Figure 1, "David Johnson," "Maine," "September 18," and "Main Hospital" are considered sensitive entities, correspondingly categorized into PII, geopolitical entities (GPE) (i.e., countries, cities, and states), time, and organization names. The first and second sentences consisting of the sensitive entities are considered sensitive sentences. Meanwhile, the third and fourth sentences are considered non-sensitive sentences, because they do not contain any sensitive entities.

Privacy Risk. It is well-known that trained ML model parameters can disclose information about training data (Carlini et al., 2020; Dwork, 2008), especially in NLMs (McMahan et al., 2017; Carlini

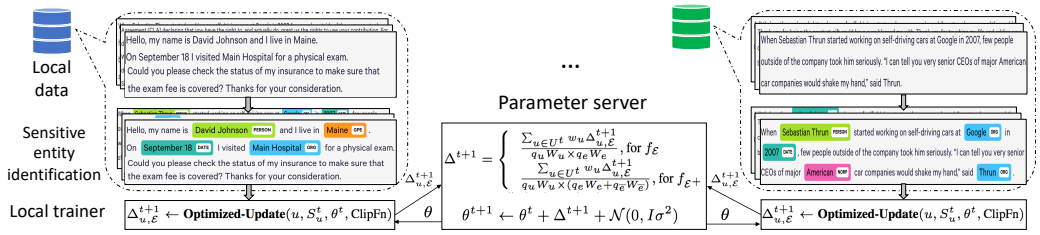


Figure 1: User-Entity DP. Data from users is processed to identify sensitive entities, before being trained with local trainers. Bounded gradients from local trainers are aggregated at a server with additive noise. Updated model are sent back to local trainers for next rounds.

et al., 2020). Given a data sample and model parameters, by using a membership inference attack (Shokri et al., 2017; Salem et al., 2018; Yeom et al., 2018), adversaries can infer whether the training used the sample or not. In NLMs, by using training data extracting attacks (Carlini et al., 2020), adversaries can accurately recover individual training examples, such as full names, email addresses, and phone numbers of individuals. Access to these can lead to severe privacy breaches.

Differential Privacy. To avoid these privacy risks, DP guarantees restriction of the adversaries in what they can learn from the training data given the model parameters by ensuring similar model outcomes with and without any single training sample. Let us revisit the definition of DP, as follows:

Definition 1. *(ϵ, δ)-DP (Dwork et al., 2006). A randomized algorithm $\mathcal{A}$ fulfills (ϵ, δ)-DP, if for any two adjacent datasets D and D' differing by at most one sample, and for all outcomes $\mathcal{O} \subseteq Range(A)$:*

$$Pr[\mathcal{A}(D) = \mathcal{O}] \leq e^{\epsilon} Pr[\mathcal{A}(D') = \mathcal{O}] + \delta \tag{1}$$

with a privacy budget ϵ and a broken probability δ.

The privacy budget ϵ controls the amount by which the distributions induced by D and D' may differ. A smaller ϵ enforces a stronger privacy guarantee. The broken probability δ means the highly unlikely "bad" events, in which an adversary can infer whether a particular data sample belongs to the training data, happen with the probability $\leq \delta$.

There are different levels of DP protection in literature categorized into four research lines, including sample-level DP, user-level DP, element-level DP, and local (feature-level) DP. Let us revisit these DP levels and distinguish them with our goal.

Sample-level DP. Traditional DP mechanisms (Roth, 2012; Dwork et al., 2014; Wu et al., 2017; Bassily et al., 2014; Pan et al., 2020) ensure DP at the sample-level, in which adjacent datasets D and D' are different from at most a single training sample. Sample-level DP does not protect privacy for users. That is different from our goal. We aim at protecting privacy for users and sensitive entities, which are different from data samples.

User-level DP. To protect privacy for users, who may contribute more than one training sample, rather than a single sample, McMahan et al. (2017) proposed a user-level DP, in which neighboring databases D and D' are defined to be different from all of the samples associated with an arbitrary user in the training set. Several works follow this direction (Kairouz et al., 2019; Ramaswamy et al., 2020). User-level DP is different from our goal, since it has not been designed to guarantee privacy for sensitive entities in the training set.

Element-level DP. Asi et al. (2019) introduce element-level DP, in which users are partitioned based on sensitive elements, which will be protected in a way that an adversary cannot infer whether a user has a sensitive element in her/his data, e.g., if a user has ever sent a curse word in his/her messages or not. Similar to sample-level DP, element-level DP is different from our goal, since it does not provide DP protection for users.

Local (Feature-level) DP. Lyu et al. (2020a) proposed a notion of word-level local DP for a sentence's embedding features, in which two adjacent sentences x and x' are different at most one word:

$$Pr[\mathcal{A}(f(x)) = \mathcal{O}] \leq e^{\epsilon} Pr[\mathcal{A}(f(x')) = \mathcal{O}] \tag{2}$$

where $f(x)$ extracts embedding features of x and $\mathcal{A}$ is a randomized algorithm, such as a Laplace mechanism (Dwork et al., 2014). In a similar effort, Lyu et al. (2020b) applied a randomized response mechanism (Erlingsson et al., 2014; Bassily & Smith, 2015; Wang et al., 2017) on top of

binary encoding of embedded features' real values to achieve local DP feature embedding. The approaches proposed in (Lyu et al., 2020a;b) are different from our goal, since they do not offer user-level privacy protection. In addition, the binary encoding in (Lyu et al., 2020b) consumes large privacy budgets compared with our approaches (**Appendix H**) and the DPNR does NOT offer word-level DP (**Appendix G**) (confirmed by the authors Lyu et al.).

3 USER-ENTITY DIFFERENTIAL PRIVACY

To preserve privacy for both users and sensitive entities in NLMs, we propose a new definition of user-entity adjacent databases, as follows: Two databases D and D' are user-entity adjacent if they differ in a single user and a single sensitive entity; that is, if one user u' and one sensitive entity e' are present in one database (i.e., D') and are absent in the other (i.e, D). Together with the absence of all data samples from the user u' in D, all data samples (across users) consisting of the sensitive entity e' also absent in D. This is because one user can have multiple sentences (samples), and one sensitive entity can exist in multiple sentences for training. The definition of our user-entity adjacent databases is presented as follows:

Definition 2. *User-Entity Adjacent Databases. Two databases D and D' are called user-entity adjacent if: $\|U - U'\|_1 \leq 1$ and $\|E - E'\|_1 \leq 1$, where U and E are the sets of users and sensitive entities in D, and U' and E' are the sets of users and sensitive entities in D'.*

Given the user-entity adjacent datasets, we present our UeDP in the following definition.

Definition 3. *(ϵ, δ)-UeDP. A randomized algorithm A is (ϵ, δ)-UeDP if for all outcomes $O \subseteq Range(A)$ and for all user-entity adjacent databases D and D', we have:*

$$Pr[A(D) = O] \leq e^{\epsilon} Pr[A(D') = O] + \delta \tag{3}$$

with a privacy budget ϵ and a broken probability δ.

If a training set does not have sensitive entity indicators, we suggest several ways to identify sensitive entities in textual data, such as: 1) Using Named Entity Recognition (NER) datasets (Sang & De Meulder, 2003; Grishman & Sundheim, 1996; Weischedel et al., 2011; Balasuriya et al., 2009; Derczynski et al., 2017; Liu & Lane, 2017; Lim et al., 2017; Stubbs et al., 2015); and 2) For textual datasets that do not have NER labels or sensitive entity indicators, there are publicly available tool-kits for detecting named entities or PII in text. For instance, Spacy (Honnibal & Montani, 2017), Stanza (Qi et al., 2020), and Microsoft Presidio. These approaches and tool-kits are user-friendly in identifying sensitive entities, making our UeDP practical. Please refer to **Appendix B** for details.

4 PRESERVING UEDP IN NLMS

To preserve UeDP, we focus on answering two questions: (1) How to bound the sensitivity of an NLM under UeDP; and (2) How to address the trade-off between privacy loss and model utility?

Overview. To answer these questions, we present a novel algorithm, called UeDP-Alg (Figure 1, Alg. 1 in **Appendix C**), in which we first sample a set of users and then sample a set of sensitive entities at each training step. Then, the set of sensitive entities are used to identify sensitive samples (sentences) consisting of each of the sampled users. A bounded gradient for each sampled user is computed using these sensitive samples. Next, we develop a new estimator to aggregate bounded gradients from all the users with an additive Gaussian noise enabling large-step model updates under UeDP protection. Finally, we optimize the trade-off between privacy loss and model utility by considering non-sensitive training data to tighten the user-entity sensitivity bound. We discover that the lower the numbers of sensitive samples and the greater the numbers of non-sensitive samples in the training set, the smaller amount of noise is injected into our model. That offers better model utility under the same UeDP protection.

4.1 UEDP PRESERVING ALGORITHM

Our UeDP-Alg takes the dataset D containing a set of users U and a set of sensitive entities E, and hyper-parameters as inputs. At each iteration t, we randomly sample U^t users from U and E^t sensitive entities from E, with sampling rates q_u and q_e, respectively (Lines 8 and 10). Then, we use all sensitive samples consisting of the sensitive entities in E^t belonging to the selected users in U^t for training. Like (McMahan et al., 2017), we leverage the basic federated learning setting in

(McMahan et al., 2016) to compute gradients of model parameters for a particular user, denoted as $\Delta^{t+1}_{u,\mathcal{E}}$ (Line 11). Here, we clip the per-user gradients so that its l_2-norm is bounded by a predefined gradient clipping bound β (**Simple-Update**$(\cdot)$, Lines 18 - 23). Next, a weighted-average estimator $f_{\mathcal{E}}$ is employed to compute the average gradient Δ^{t+1} using the clipped gradients $\Delta^{t+1}_{u,\mathcal{E}}$ gathered from all the selected users (Line 12). Finally, we add random Gaussian noise $\mathcal{N}(0, I\sigma^2)$ to the model update (Line 14). During the training, $\mathcal{M}$ is used to compute the T training steps' privacy budget consumption (Lines 15-16).

In this process, we need to bound the sensitivity of the weighted-average estimator $f_{\mathcal{E}}$ for per-user gradients $\Delta^{t+1}_{u,\mathcal{E}}$. We first consider the following simple estimator (Line 12, using the Simple-Update$(\cdot)$), with both sampling rates q_u for the user-level and q_e for the sensitive entity-level:

$$f_{\mathcal{E}}(U^t, E^t) = \frac{\sum_{u\in U^t} w_u \Delta^{t+1}_{u,\mathcal{E}}}{q_u W_u \times q_e W_e} \text{ s.t. } \Delta^{t+1}_{u,\mathcal{E}} = \sum_{e\in E^t} w_e \Big(\sum_{s\in S^t_{ue}} \Delta_{u,s} \Big) \tag{4}$$

where $w_u, w_e \in [0, 1]$ be weights associated with a user u and with a sensitive entity e. These weights capture the influence of a user and a sensitive entity to the model outcome. S^t_{ue} is a set of samples belonging to user u, and each of samples $s \in S^t_{ue}$ consists of the sensitive entity e. $\Delta_{u,s}$ is the parameter gradients computed using the sample s. In addition, $W_u = \sum_u w_u$ and $W_e = \sum_e w_e$.

The estimator $f_{\mathcal{E}}$ is unbiased to the sampling process; since $\mathbb{E}[\sum_{u\in U^t} w_u] = q_u W_u$ and $\mathbb{E}[\sum_{e\in E^t} w_e] = q_e W_e$. The sensitivity of the estimator $f_{\mathcal{E}}$ can be computed as: $\mathbb{S}(f_{\mathcal{E}}) = \max_{u',e'} \| f_{\mathcal{E}}(\{U^t \cup u', E^t \cup e'\}) - f_{\mathcal{E}}(\{U^t, E^t\})\|_2$, where the added user u' can have arbitrary data and e' is an arbitrary sensitive entity.

Given that $\Delta^{t+1}_{u,\mathcal{E}}$ is $l_2(\beta)$-norm bounded, where β is the radius of the norm ball by replacing $\Delta^{t+1}_{u,\mathcal{E}}$ with $\Delta^{t+1}_{u,\mathcal{E}} \cdot \min\left(1, \frac{\beta}{\|\Delta^{t+1}_{u,\mathcal{E}}\|}\right)$ (Lines 32-33), the sensitivity of the estimator $\mathbb{S}(f_{\mathcal{E}})$ is also bounded.

Lemma 1. *If for all users u we have $\|\Delta^{t+1}_{u,\mathcal{E}}\|_2 \leq \beta$, then $\mathbb{S}(f_{\mathcal{E}}) \leq \frac{(q_u|U|+1)\max(w_u)\beta}{q_u W_u \times q_e W_e}$.*

The proof of Lemma 1 is in **Appendix D**. By applying Lemma 1, given a hyper-parameter z, the noise scale σ for the estimator $f_{\mathcal{E}}$ (Line 13) is:

$$\sigma = z\mathbb{S}(f_{\mathcal{E}}) = \frac{z(q_u|U|+1)\max(w_u)\beta}{q_u W_u \times q_e W_e} \tag{5}$$

We show that this approach achieves (ϵ, δ)-UeDP, by applying the moments accountant $\mathcal{M}$ (Abadi et al., 2016) to bound the total privacy loss of T steps of the Gaussian mechanism with the noise $\mathcal{N}(0, I\sigma^2)$ in Theorem 1. However, this mechanism only uses sensitive samples to train the model ignoring a large number of non-sensitive samples. As a result, it introduces a loose sensitivity bound (Lemma 1) and affects our model utility.

4.2 Optimizing Utility - Privacy Trade-off

To improve model utility under the same UeDP protection, we incorporate non-sensitive samples into the training process. There are two critical impacts of doing so: **1)** reducing privacy loss since the model can learn more from non-sensitive samples, limiting the knowledge the model learns from sensitive samples, and **2)** deriving a notably tighter sensitivity bound. Our algorithm is as follows.

In each step t, for each user u, by using the **Optimized-Update**$(\cdot)$ (Lines 11, 24 - 31), we randomly select non-sensitive samples $\overline{S}^t_u$ from all the non-sensitive samples consisting in the user u's data $\overline{S}_u$ with the probability $q_{\overline{s}}$ (Line 25). The selected non-sensitive samples $\overline{S}^t_u$ will be merged with the sensitive samples S^t_u to compute the per-user gradients (Lines 26 - 30). Next, we also clip the per-user gradients so that its l_2-norm is bounded by β (Line 31). To tighten the sensitivity bound, we propose a new estimator $f_{\mathcal{E}^+}$ (Line 12), as follows:

$$f_{\mathcal{E}^+}(U^t, E^t) = \frac{\sum_{u\in U^t} w_u \Delta^{t+1}_{u,\mathcal{E}}}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})} \text{ s.t. } \Delta^{t+1}_{u,\mathcal{E}} = \Big(\sum_{e\in E^t} w_e \Delta_{u,e} + \sum_{s\in \overline{S}^t_u} w_s \Delta_{u,s} \Big) \tag{6}$$

where $w_s \in [0, 1]$ is a weight associated with a non-sensitive sample s to capture the influence of s to the model outcome, and $W_{\overline{s}} = \sum_{s\in\overline{S}} w_s$ given a set of non-sensitive samples $\overline{S}$ in the data D.

Since $\mathbb{E}[\sum_{e\in E^t} w_e + \sum_{s\in\overline{S}} w_s] = q_e W_e + q_{\overline{s}} W_{\overline{s}}$, the estimator $f_{\mathcal{E}^+}$ is unbiased. The sensitivity of the estimator $\mathbb{S}(f_{\mathcal{E}^+})$ is computed as:

$$\mathbb{S}(f_{\mathcal{E}^+}) = \max_{u',e'} \| f_{\mathcal{E}^+}(\{U^t \cup u', E^t \cup e'\}) - f_{\mathcal{E}^+}(\{U^t, E^t\}) \|_2$$

$\mathbb{S}(f_{\mathcal{E}^+})$ is bounded in the following lemma.

Lemma 2. *If for all users u we have $\|\Delta_{u,\mathcal{E}}^{t+1}\|_2 \leq \beta$, then $\mathbb{S}(f_{\mathcal{E}^+}) \leq \frac{(q_u|U|+1)\max(w_u)\beta}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})}$.*

The proof of Lemma 2 is in **Appendix E**. By applying Lemma 2, the noise scale σ becomes:

$$\sigma = z\mathbb{S}(f_{\mathcal{E}^+}) = \frac{z(q_u|U|+1)\max(w_u)\beta}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})} \tag{7}$$

The noise scale σ in Eq. 7 is significantly smaller than the noise scale in Eq. 5. In fact, given a fixed set of all samples, if there are more sensitive samples, we will have: **(1)** More non-sensitive samples will contribute to the gradients through $\overline{S}_u^t$; and **(2)** A smaller noise scale σ, given a larger term $q_e W_e + q_{\overline{s}} W_{\overline{s}}$ (i.e., a larger set $\overline{S}$), since we can set $q_{\overline{s}}$ to be larger than q_e. As a result, the less number of sensitive samples and the more number of non-sensitive samples in the training set, we can inject the smaller amount of noise into our model (proportionally to $q_{\overline{s}} W_{\overline{s}}$). That enables us to reduce the privacy loss while improving our model utility.

UeDP Guarantee. Given the bounded sensitivity of the estimators, moments accountant $\mathcal{M}$ (Abadi et al., 2016) is used to get a tight bound on the total UeDP privacy consumption of T steps of the Gaussian mechanism with the noise $\mathcal{N}(0, I\sigma^2)$ (Line 14), in the following theorem.

Theorem 1. *For the estimators $f_{\mathcal{E}}$ and $f_{\mathcal{E}^+}$, the moments accountant of the sampled Gaussian mechanism correctly computes the UeDP privacy loss with the scale $z = \sigma/\mathbb{S}(f_{\mathcal{E}})$ for $f_{\mathcal{E}}$ and $z = \sigma/\mathbb{S}(f_{\mathcal{E}^+})$ for $f_{\mathcal{E}^+}$ for T training steps.*

The proof of Theorem 1 is in **Appendix F**.

5 Experimental Results

We conducted an extensive experiment, both in theory and on benchmark datasets, to shed light on understanding 1) the integrity of sensitive entity identification, 2) the interplay among the UeDP privacy budget (ϵ, δ), different types of sensitive entities (i.e., organization, location, PII, and miscellaneous entities), and model utility, and 3) whether considering non-sensitive samples will improve our model utility under the same UeDP protection.

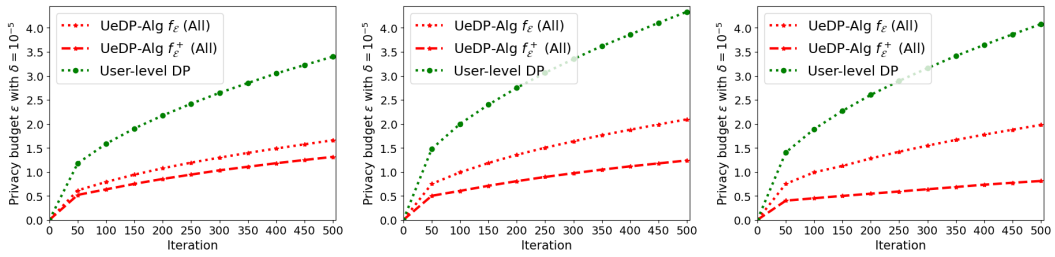
Baseline Approaches. We evaluate our **UeDP-Alg** (with estimators $f_{\mathcal{E}}$ and $f_{\mathcal{E}^+}$) in comparison with both noiseless and privacy-preserving mechanisms (either user level or entity level), including: **(1) User-level DP** (McMahan et al., 2017), which is the state-of-the-art DP-preserving model closely related to our work; **(2) De-Identification** (Dernoncourt et al., 2017), which is considered a strong baseline to provide privacy protection to sensitive entities. Although sensitive entities are masked to hide them in the training process, De-Identification does not offer formal privacy protection to either the data owners or sensitive entities; and **(3)** A **Noiseless** model, which is a language model trained without any privacy-preserving mechanisms. In our experiment, our algorithms and baseline approaches, i.e., UeDP-Alg, User-level DP, and De-Identification, are applied on the noiseless model in the training process. As in our literature review, there are no other appropriate DP-preserving baselines to UeDP protection, i.e., (Asi et al., 2019; Lyu et al., 2020a;b).

Evaluation Tasks and Metrics. Two tasks are considered in our experiment: **(1)** next word prediction and **(2)** text classification. For the next word prediction, we employ the widely used perplexity (Mikolov et al., 2011b; Bengio et al., 2003). Perplexity is the exponential of the average negative log-likelihood to measure how well a language model predicts a word or a sequence of words. The smaller perplexity is, the better model is. For the text classification, we use the test error rate as in earlier work (Howard & Ruder, 2018). Test error rate implies prediction error on a test set, so it is likely 1 - the test set's accuracy. The lower the test error rate is, the better model is.

Data and Model Configuration. For the reproducibility sake, all details about our datasets, data processing, and model configuration are included in **Appendices I and J**. We carried out our experiment on three textual datasets, including the CONLL-2003 news dataset (Sang & De Meulder,

Table 1: Breakdown of CONLL-2003, AG, and SEC datasets.

Dataset	$\|\mathcal{V}\|$	# of samples	# of users	# of sensitive samples				
				Org	Loc	Person	Misc	All
CONLL-2003	8,882	14,040	946	5,187	5,433	4,406	3,438	11,176
				Org	Loc	GPE	PII	All
AG	30,000	112,000	7,536	58,177	39,988	18,506	42,683	67,157
				Org	Loc	GPE	PII	All
SEC	12,651	5,188	1,592	1,955	273	60	357	2,166



(a) CONLL-2003-all entities (b) AG-all entities (c) SEC-all entities

Figure 2: Privacy budget of UeDP-Alg $f_{\mathcal{E}}$, UeDP-Alg $f_{\mathcal{E}+}$, and User-level DP as a function of iterations in CONLL-2003, AG, and SEC datasets. "All" mean all sensitive entity types are used in training. (The lower the better)

2003), AG's corpus of news articles, and our collected Security and Exchange Commision (SEC) financial contract dataset. The data breakdown for these datasets is in Table 1.

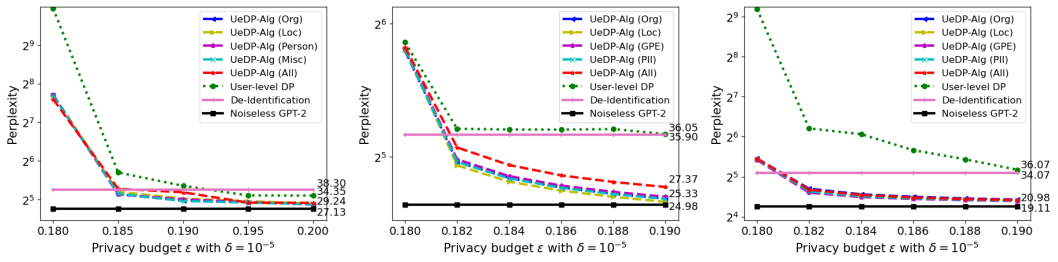
For the next word prediction, we employ a GPT-2 model (Radford et al., 2019), which is one of the state-of-art text generation models (Radford et al., 2018; Brown et al., 2020). To make the work easily reproducible, we use a version of the pretrained GPT-2 that has 12-layer, 768-hidden, 12-heads, 117M parameters, and then fine-tune with the aforementioned datasets as our Noiseless GPT-2 model. For the text classification, we fine-tune a Noiseless BERT (i.e., BERT-Base-Uncased) pre-trained model (ber; Devlin et al., 2018) that has 12-layer, 768-hidden, 12-heads, and 110M parameters with adding a softmax function on top of the BERT model. To test the effectiveness and adaptability of our mechanism across models, we also conducted experiments with an AWD-LSTM model (Merity et al., 2017; Merity, 2019) (ASGD Weight-Dropped LSTM), which has a much fewer parameters compared with GPT-2 and BERT. In AWD-LSTM model, we use a three-layer LSTM model with $1,150$ units in the hidden layer and an embedding input layer of size 100.

Evaluation Results. In order to answer our evaluation questions, we conducted the following comparisons: **(1)** examining the sensitive information coverage of sensitive entities identified by the sensitive entity identification, i.e., spaCy (Honnibal & Montani, 2017), **(2)** estimators $f_{\mathcal{E}}$, $f_{\mathcal{E}+}$, and User-level DP; **(3)** the interplay between privacy budget and model utility; **(4)** the impacts of different sensitive entity categories and non-sensitive sentences on the privacy budget and model utility; and **(5)** confirming our results in the text classification task. Our result analysis is as follows:

• **Integrity of sensitive entities.** In practice, sensitive information does not have to be some entities. We can consider sensitive entities as a sub-type of sensitive information and vice-versa. However, identifying sensitive information is out of the scope of our work. Note that our mechanism is not limited to sensitive entities, since we can naturally extend it to cover sensitive information. For instance, we can treat sensitive information as a category of sensitive entities when a sensitive information identification method is available.

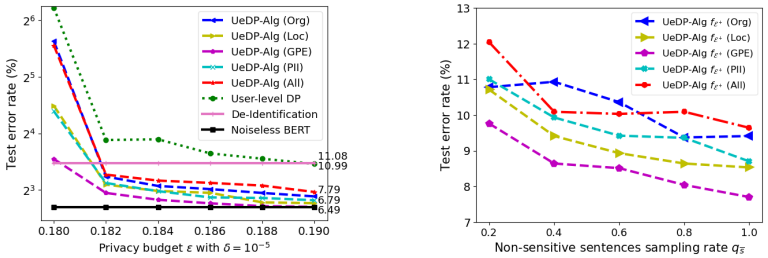
In our work, we utilize spaCy (Honnibal & Montani, 2017), which is one of the state-of-the-art large-scale entity recognition systems, to identify sensitive entities. In order to evaluate the integrity of identified sensitive entities, we conducted a clarification on Amazon Mechanical Turk (AMT), and we found that the results from spaCy covers over 94% of sensitive information as identified by AMT workers. Note that we recruited master-level AMT workers for a high quality of results, and we provided detailed guidance before AMT workers conducted the task. Each data sample was assigned to 3 master-level workers to mitigate bias and subjective views. Consequently, our experiments using the spaCy identified sensitive entities are solid.

• **Comparing Estimators $f_{\mathcal{E}}$, $f_{\mathcal{E}+}$, and User-level DP.** In this analysis, we set $q_u = 0.05$, $q_e = 0.5$, $q_{\bar{s}} = 1$, $z = 2$, and compute privacy budget ϵ at $\delta = 10^{-5}$ (a typical value of δ in DP) as a function



(a) CONLL-2003 dataset (b) AG dataset (c) SEC dataset

Figure 3: Next word prediction results using the GPT-2 model. (The lower the better)



(a) AG dataset (b) AG dataset with varying $q_{\bar{s}}$

Figure 4: Text classification results on the AG dataset using the BERT model. With $q_{\bar{s}} = 0.0$, the test error rate is 75% in all cases. (The lower the better)

of the training steps T. Figure 2 shows curves of using different estimators and the User-level DP with all entities in CONLL-2003, AG, and SEC datasets. More results with a breakdown of sensitive entities are in Figure 6 (**Appendix K**).

Our UeDP-Alg with $f_{\mathcal{E}^+}$ achieves a notably tighter privacy budget compared with $f_{\mathcal{E}}$ and with the User-level DP mechanism in all scenarios in CONLL-2003, AG, and SEC datasets. User-level DP consumes a much higher privacy budget ϵ compared with both of our estimators $f_{\mathcal{E}}$ and $f_{\mathcal{E}^+}$. For instance, at $T = 50$, the values of ϵ in organization entities of $f_{\mathcal{E}}$ and $f_{\mathcal{E}^+}$, and the value of ϵ of the User-level DP in: **(1)** the CONLL-2003 dataset are 0.62, 0.36, and 1.18; **(2)** the AG dataset are 0.75, 0.47, and 1.48; and **(3)** the SEC dataset are 0.71, 0.38, and 1.40 respectively.

Significantly, the privacy budget (ϵ) gap between User-level DP, $f_{\mathcal{E}}$, and $f_{\mathcal{E}^+}$ is proportionally increased to the number of steps T. That means the more training steps T, the larger the privacy budget our model can save compared with the baseline User-level DP. That is a promising result in the context that our model provides DP protection for both users and sensitive entities, compared with only protection for users in User-level DP. We observe a similar phenomenon on all entities and other sensitive categories (**Appendix K**).

• **Privacy Budget (ϵ, δ)-UeDP and Model Utility.** From our theoretical analysis, $f_{\mathcal{E}^+}$ is better than the estimator $f_{\mathcal{E}}$. Therefore, for the sake of simplicity, we only consider UeDP-Alg $f_{\mathcal{E}^+}$ instead of showing results from both estimators. From now, UeDP-Alg is used to indicate the use of our estimator $f_{\mathcal{E}^+}$. Figure 3 illustrates the perplexity as a function of the privacy budget ϵ for an GPT-2 model trained on a variety of sensitive entity categories in UeDP, User-level DP, and De-Identification. The noiseless GPT-2 (for the next word prediction) and BERT (for the text classification) models are considered an upper-bound performance mechanism without offering any privacy protection.

In the CONLL-2003 dataset (Figure 3a), there are NER labels for person, location, organization, and miscellaneous entities; therefore, we choose these types as sensitive entity categories to protect in UeDP-Alg. UeDP-Alg achieves a better perplexity compared with User-level DP under a tight privacy budget $\epsilon \in [0.18, 0.20]$. Also, from $\epsilon = 0.185$ (a tight privacy protection), our UeDP-Alg achieves a better perplexity than De-Identification. In fact, at at $\epsilon = 0.185$, our UeDP-Alg achieves 35.09 for person, 35.34 for organization, 35.57 for miscellaneous, 36.79 for location entities, compared with 52.01 in User-level DP. When spending more privacy budget ($\epsilon \geq 0.195$), both UeDP-Alg and User-level DP converge at a very competitive perplexity level, approaching the upper-bound Noiseless GPT-2. For instance, at $\epsilon = 0.20$, there are significant perplexity drops given UeDP-Alg and User-level DP mechanisms, i.e., our UeDP-Alg is 29.24 for person, 29.35 for mis-

cellaneous, 29.58 for organization, and 29.75 for location entities. Meanwhile, the perplexity values of User-level DP, De-Identification, and the noiseless GPT-2 model are 30.15, 38.30, and 27.13.

Results on AG and SEC datasets (Figures 3b and 3c) further strengthen our observations. In AG and SEC datasets, we applied spaCy to identify different sensitive entity categories, such as GPE, location, organization, and PII (i.e., person and location information). UeDP-Alg achieves better results compared with User-level DP in all considering sensitive entity categories and privacy budgets, and outperforms De-Identification in most cases. That is promising and consistent with our previous analysis. For instance, in the AG dataset, at $\epsilon = 0.19$, our UeDP-Alg achieves 25.33 for location, 25.72 for PII, 25.77 for organization, and 26.01 for GPE entities, compared with 36.05 in User-level DP. De-Identification obtains 35.90, and the upper bound result in the noiseless GPT-2 model is 24.98. Similarly, in the SEC dataset (Figure 3c), at $\epsilon = 0.19$, UeDP-Alg achieves perplexity of 20.98 in GPE, 21.12 in PII, 21.22 in location, 21.50 in organization, and 21.33 in all entities, compared with 36.07 in User-level DP, and 34.07 in De-Identification. In AG and SEC datasets, at a tight privacy budget, i.e., $\epsilon = 0.19$, our UeDP-Alg has better perplexity values than the De-Identification, approaching the noiseless GPT-2 model.

• **Sensitive Entity Categories.** In all datasets (Figures 3 and 6, **Appendix K**), *the more sensitive data samples to protect, the higher the privacy budget is needed and the lower performance of the model achieves* (i.e., higher values of perplexity). These theoretical and experimental results are consistent with our theoretical analysis after Lemma 2. For instance, in the SEC dataset, the number of sensitive samples in each category is as follows: 60 in GPE, 273 in location, 357 in PII, $1,955$ in organization, and $2,166$ in all entities. After 500 steps, the respective values of ϵ are 0.19 in GPE, 0.24 in location, 0.26 in PII, 0.73 in organization, 0.81 in all entities, and 4.08 in User-level DP (Figure 6). At $\epsilon = 0.18$ (Figure 3c), we obtain perplexity values of 42.63 in GPE, 43.21 in location, 43.30 in PII, 43.70 in organization, 43.77 in all entities, and a 583.06 in User-level DP.

• **Text classification.** Figure 4a shows that our UeDP-Alg achieves lower test error rates in terms of text classification on the AG dataset than baseline approaches in most cases across different types of sensitive entities under a very tight UeDP protection ($\epsilon \in [0.18, 0.19]$). This is a promising result. When ϵ is higher, the test error rates of both UeDP-Alg and User-level DP drop, approaching the noiseless BERT model's upper-bound result.

• **Non-Sensitive Sentences.** To shed light into the impact the non-sensitive sentence sampling rate $q_{\bar{s}}$ on model utility under UeDP protection, we varied the value of $q_{\bar{s}}$ from 0 to 1 in all datasets and tasks. Figures 4b, 7, 8, and 10b (**Appendix K**) show that considering non-sensitive sentences (i.e., $q_{\bar{s}} > 0$) significantly improves model utility (i.e., perplexity or test error rate) compared with only considering sensitive-sentences (i.e., $q_{\bar{s}} = 0$). However, different tasks on different datasets may have different optimal values of $q_{\bar{s}}$. This opens a new research question on how to theoretically approximate the optimal value of $q_{\bar{s}}$.

Results on the AWD-LSTM model (Figures 8-10 in **Appendix K**) further strengthen our observations. In our experiments, the AWD-LSTM model generally obtains comparable results with GPT-2 (for next word prediction) and BERT (for text classification) at a higher privacy budget range (i.e., $\epsilon \in [0.5, 3.0]$ in the AWD-LSTM model compared with $\epsilon \in [0.18, 0.2]$ in the GPT-2 and BERT models). This is because the GPT-2 and BERT models are pretrained on large-scale datasets, so that it is easily adapted to the idiosyncrasies of a target task (i.e., next word prediction or text classification) compared with the AWD-LSTM model trained from scratch.

6 Conclusion

In this paper, we developed a novel notion of user-entity DP (UeDP), providing protection to both the participation information of users and sensitive entities in learning NLMs. That is one step forward compared with existing protection levels in DP. By incorporating non-sensitive samples in the training process, we addressed the trade-off between model utility and privacy loss with a tight bound of model sensitivity. Theoretical analysis and rigorous experiments conducted on real-world datasets shown that our UeDP-Alg outperforms baseline approaches in fundamental NLM tasks, i.e., next word prediction and text classification, under rigorous UeDP protection, i.e., small privacy budget ϵ. In addition, considering non-sensitive samples into our UeDP estimators notably improves model utility under the same UeDP protection. The more number of sensitive entities is, the lower the model utility will be; and vice-versa.

REFERENCES

Google ai. pre-trained bert model. https://bert-as-service.readthedocs.io/en/latest/section/get-start.html#installation.

M. Abadi, A. Chu, I. Goodfellow, H.B. McMahan, I. Mironov, K. Talwar, and L. Zhang. Deep learning with differential privacy. In *ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, 2016.

AG's corpus of news articles. http://groups.di.unipi.it/~gulli/AG_corpus_of_news_articles.html. Accessed: 2021-02-03.

A. Al Badawi, L. Hoang, C.F. Mun, K. Laine, and K.M.M. Aung. Privft: Private and fast text classification with homomorphic encryption. *IEEE Access*, 8:226544–226556, 2020.

A. Alnemari, C.J. Romanowski, and R.K. Raj. An adaptive differential privacy algorithm for range queries over healthcare data. In *2017 IEEE International Conference on Healthcare Informatics (ICHI)*, pp. 397–402, 2017.

H. Asi, J. Duchi, and O. Javidbakht. Element level differential privacy: The right granularity of privacy. *arXiv preprint arXiv:1912.04042*, 2019.

E. Bagdasaryan, O. Poursaeed, and V. Shmatikov. Differential privacy has disparate impact on model accuracy. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 15479–15488, 2019.

D. Balasuriya, N. Ringland, J. Nothman, T. Murphy, and J.R. Curran. Named entity recognition in wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pp. 10–18, 2009.

R. Bassily and A. Smith. Local, private, efficient protocols for succinct histograms. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pp. 127–135, 2015.

R. Bassily, A. Smith, and A. Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 464–473, 2014.

Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.

N. Carlini, F. Tramer, E. Wallace, M. Jagielski, A. Herbert-Voss, K. Lee, A. Roberts, T. Brown, D. Song, U. Erlingsson, et al. Extracting training data from large language models. *arXiv preprint arXiv:2012.07805*, 2020.

Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *CoRR*, abs/1802.08232, 2018. URL http://arxiv.org/abs/1802.08232.

Mahawaga Arachchige Pathum Chamikara, Peter Bertók, Ibrahim Khalil, Dongxi Liu, and Seyit Camtepe. Local differential privacy for deep learning. *CoRR*, abs/1908.02997, 2019. URL http://arxiv.org/abs/1908.02997.

S.F. Chen, D. Beeferman, and R. Rosenfeld. Evaluation metrics for language models. 1998.

Cometomyhead academic news search engine. http://newsengine.di.unipi.it/. Accessed: 2021-02-03.

L. Derczynski, E. Nichols, M. van Erp, and N. Limsopatham. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pp. 140–147, 2017.

F. Dernoncourt, J.Y. Lee, O. Uzuner, and P. Szolovits. De-identification of patient notes with recurrent neural networks. *Journal of the American Medical Informatics Association*, 24(3):596–606, 2017.

J. Devlin, M. W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

John C Duchi, Michael I Jordan, and Martin J Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pp. 429–438, 2013.

C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference*, pp. 265–284, 2006.

C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3-4):211–407, 2014.

Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pp. 1–19. Springer, 2008.

Cynthia Dwork and Jing Lei. Differential privacy and robust statistics. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing*, STOC '09, pp. 371–380, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585062. doi: 10.1145/1536414.1536466. URL https://doi.org/10.1145/1536414.1536466.

U. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 1054–1067, 2014.

L. Fan. Image pixelization with differential privacy. In *IFIP Annual Conference on Data and Applications Security and Privacy*, pp. 148–162, 2018.

R. Grishman and B.M. Sundheim. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

M. Honnibal and I. Montani. Spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1), 2017.

J. Howard and S. Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *arXiv preprint arXiv:1912.04977*, 2019.

H.B. Kartal, X. Liu, and X.B. Li. Differential privacy for the vast majority. *ACM Transactions on Management Information Systems (TMIS)*, 10(2):1–15, 2019.

J. Lee and D. Kifer. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1656–1665, 2018.

N. Li, W. Qardaji, and D. Su. On sampling, anonymization, and differential privacy or, k-anonymization meets differential privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, pp. 32–33, 2012.

S.K. Lim, A.O. Muis, W. Lu, and C.H. Ong. Malwaretextdb: A database for annotated malware articles. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1557–1567, 2017.

B. Liu and I. Lane. Multi-domain adversarial learning for slot filling in spoken language understanding. *arXiv preprint arXiv:1711.11310*, 2017.

L. Lyu, X. He, and Y. Li. Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. *arXiv preprint arXiv:2010.01285*, 2020a.

L. Lyu, Y. Li, X. He, and T. Xiao. Towards differentially private text representations. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1813–1816, 2020b.

Lingjuan Lyu, Yitong Li, Xuanli He, and Tong Xiao. Towards differentially private text representations. In *Proceedings of the SIGIR'20*, pp. 1813–1816, 2020c. doi: 10.1145/3397271.3401260.

H.B. McMahan, E. Moore, D. Ramage, and B.A. y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

H.B. McMahan, D. Ramage, K. Talwar, and L. Zhang. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*, 2017.

S. Merity. Single headed attention rnn: Stop thinking with your head. *arXiv preprint arXiv:1911.11423*, 2019.

S. Merity, N. S. Keskar, and R. Socher. Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182*, 2017.

T. Mikolov, A. Deoras, S. Kombrink, L. Burget, and J. Černockỳ. Empirical evaluation and combination of advanced language modeling techniques. In *International Speech Communication Association*, 2011a.

T. Mikolov, S. Kombrink, L. Burget, J. Černockỳ, and S. Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5528–5531, 2011b.

I. Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, 2017.

L. Ou, Z. Qin, S. Liao, Y. Hong, and X. Jia. Releasing correlated trajectories: Towards high utility and optimal differential privacy. *IEEE Transactions on Dependable and Secure Computing*, 2018.

X. Pan, M. Zhang, S. Ji, and M. Yang. Privacy risks of general-purpose language models. In *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1314–1331, 2020.

N.H. Phan, Y. Wang, X. Wu, and D. Dou. Differential privacy preservation for deep auto-encoders: an application of human behavior prediction. In *AIII*, volume 16, pp. 1309–1316, 2016.

N.H. Phan, M. Vu, Y. Liu, R. Jin, D. Dou, X. Wu, and M.T. Thai. Heterogeneous gaussian mechanism: Preserving differential privacy in deep learning with provable robustness. *arXiv preprint arXiv:1906.01444*, 2019.

P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C.D. Manning. Stanza: A python natural language processing toolkit for many human languages. *arXiv preprint arXiv:2003.07082*, 2020.

A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever. Improving language understanding by generative pre-training. 2018.

A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

S. Ramaswamy, O. Thakkar, R. Mathews, G. Andrew, H.B. McMahan, and F. Beaufays. Training production language models without memorizing user data. *arXiv preprint arXiv:2009.10031*, 2020.

A. Roth. Buying private data at auction: the sensitive surveyor's problem. *ACM SIGecom Exchanges*, 11(1):1–8, 2012.

A. Salem, Y. Zhang, M. Humbert, P. Berrang, M. Fritz, and M. Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. *arXiv preprint arXiv:1806.01246*, 2018.

E.F. Sang and F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

R. Shokri and V. Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1310–1321, 2015.

R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 3–18, 2017.

A. Stubbs, C. Kotfila, and O. Uzuner. Automated systems for the de-identification of longitudinal clinical narratives: Overview of 2014 i2b2/uthealth shared task track 1. *Journal of Biomedical Informatics*, 58:S11–S19, 2015.

L. Sun and L. Lyu. Federated model distillation with noise-free differential privacy. *arXiv preprint arXiv:2009.05537*, 2020.

S. Wang and R.O. Sinnott. Protecting personal trajectories of social media users through differential privacy. *Computers & Security*, 67:142–163, 2017.

T. Wang, J. Blocki, N. Li, and S. Jha. Locally differentially private protocols for frequency estimation. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pp. 729–745, 2017.

R. Weischedel, S. Pradhan, L. Ramshaw, M. Palmer, N. Xue, M. Marcus, A. Taylor, C. Greenberg, E. Hovy, R. Belvin, et al. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*, 2011.

N. Wu, F. Farokhi, D. Smith, and Mohamed A. K. The value of collaboration in convex machine learning with differential privacy. *arXiv preprint arXiv:1906.09679*, 2019.

X. Wu, F. Li, A. Kumar, K. Chaudhuri, S. Jha, and J. Naughton. Bolt-on differential privacy for scalable stochastic gradient descent-based analytics. In *ACM International Conference on Management of Data*, pp. 1307–1322, 2017.

S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *2018 IEEE 31st Computer Security Foundations Symposium (CSF)*, pp. 268–282, 2018.

L. Yu, L. Liu, C. Pu, M.E. Gursoy, and S. Truex. Differentially private model publishing for deep learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 332–349, 2019.

M.T. Zia, M.A. Khan, and H. El-Sayed. Application of differential privacy approach in healthcare data–a case study. In *2020 14th International Conference on Innovations in Information Technology (IIT)*, pp. 35–39, 2020.

A NOTATIONS AND TERMINOLOGIES

Table 2: Notations and Terminologies.

Notations	Meaning
D, U, x	Dataset D consisting U users and x samples
u, n_u	Set of users U, $u \in U$, n_u is the number of samples in u
E, e	Set of sensitive entities E, $e \in E$
S, s	Set of samples having sensitive entities in E, $s \in S$
$\overline{S}$	Set of non-sensitive samples that do not consist of any sensitive entities in E
$PP = 2^{-\sum_{x \in D} p(x) \log_2 p(x)}$	Perplexity where $p(x)$ is a probability to predict the next word in x
ϵ, δ	Privacy budget ϵ and broken probability δ
D, D'	User-entity adjacent databases
$\|U - U'\|_1$	l_1 distance between two sets of users $U \in D$ and $U' \in D'$
$\|E - E'\|_1$	l_1 distance between two sets of sensitive entities $E \in D$ and $E' \in D'$
$U^t, \overline{S}^t, E^t, S^t$	Selected sets of users, non-sensitive samples, sensitive entities, sensitive samples having $e \in E^t$, with sample rates $q_u, q_{\overline{s}}, q_e$
$\mathcal{M}$	Moments accountant
$w_u = \min(\frac{n_u}{\bar{w}_u}, 1), W_u = \sum_{u \in U} w_u$	User-level weight associated with a user
$w_e = \min(\frac{n_e}{\bar{w}_e}, 1), W_e = \sum_{e \in E} w_e$	Entity-level weight associated with a sensitive sample
$w_s = \min(\frac{n_s}{\bar{w}_s}, 1), W_{\overline{s}} = \sum_{s \in \overline{S}} w_s$	Sample-level weight associated with a non-sensitive sample
$\Delta_{u,s}$	Gradients computed using the sample s of user u
β	Radius of the norm ball to bound l_2-norm of gradients
$\Delta^{t+1}_{u,\mathcal{E}} = \sum_{e \in E^t} w_e (\sum_{s \in S^t_{ue}} \Delta_{u,s})$	Updated gradients of model parameters in Simple-Update UeDP
$f_{\mathcal{E}}(U^t, E^t) = \frac{\sum_{u \in U^t} w_u \Delta^{t+1}_{u,\mathcal{E}}}{q_u W_u \times q_e W_e}$	Simple-Update UeDP at iteration t
$\mathbb{S}(f_{\mathcal{E}}) \leq \frac{(\lvert U \rvert + 1) \max(w_u) \beta}{q_u W_u \times q_e W_e}$	Sensitivity of the estimator $f_{\mathcal{E}}$
$\sigma = z\mathbb{S}(f_{\mathcal{E}}) = \frac{z(\lvert U \rvert + 1) \max(w_u) \beta}{q_u W_u \times q_e W_e}$	Noise scale σ computed for the estimator $f_{\mathcal{E}}$, z is a hyper-parameter
$\Delta^{t+1}_{u,\mathcal{E}} = (\sum_{e \in E^t} w_e \Delta_{u,e} + \sum_{s \in \overline{S}^t_u} w_s \Delta_{u,s})$	Updated gradients of model parameters in Optimized-Update UeDP
$f_{\mathcal{E}+}(U^t, E^t) = \frac{\sum_{u \in U^t} w_u \Delta^{t+1}_{u,\mathcal{E}}}{q_u W_u \times q_e W_e + q_{\overline{s}} W_{\overline{s}}}$	Optimized-Update UeDP at iteration t
$\mathbb{S}(f_{\mathcal{E}}) \leq \frac{(\lvert U \rvert + 1) \max(w_u) \beta}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})}$	Sensitivity of the estimator $f_{\mathcal{E}+}$
$\sigma = z\mathbb{S}(f_{\mathcal{E}+}) = \frac{z(\lvert U \rvert + 1) \max(w_u) \beta}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})}$	Noise scale σ computed for the estimator $f_{\mathcal{E}+}$
$\theta^{t+1} \leftarrow \theta^t + \Delta^{t+1} + \mathcal{N}(0, I\sigma^2)$	Update model parameters

B CATEGORIES OF SENSITIVE ENTITIES, AND TOOL-KITS

If a training set does not have sensitive entity indicators, we suggest several ways to identify sensitive entities in textual data, as follows:

Using Named Entity Recognition (NER) datasets. NER datasets (Sang & De Meulder, 2003; Grishman & Sundheim, 1996; Weischedel et al., 2011; Balasuriya et al., 2009; Derczynski et al., 2017; Liu & Lane, 2017; Lim et al., 2017; Stubbs et al., 2015) refer to textual data in which entities in a text are labeled based on several predefined categories. NER typically makes it easy for individuals and systems to identify and understand the subject of the given text quickly. Therefore, extracted entities are critical and should be protected. For instance, in the CONLL-2003 dataset (Sang & De Meulder, 2003), there are four entity types, i.e., location, person, organization, and miscellaneous.

Using Publicly Available Tool-kits. For textual datasets that do not have NER labels or sensitive entity indicators, there are publicly available tool-kits for detecting named entities or PII in text, for example, Spacy (Honnibal & Montani, 2017), Stanza (Qi et al., 2020), and Microsoft Presidio[1]. Spacy and Stanza deploy pre-trained NER models based on statistical learning methods to identify eighteen categories of named entities, including person, nationality or religious groups, facility, etc. (Table 3). Microsoft Presidio is another toolbox for PII detectors and NER models based on Spacy and regular expression[2]. For instance, Spacy is used as a sensitive entity identification in Figure 1 to detect "David Johnson" a person entity, "Main" a GPE entity, "September 18" a date entity, and "Main Hospital" an organization entity.

[1] https://microsoft.github.io/presidio/

[2] https://github.com/google/re2/

We present descriptions of different sensitive entity categories in the CONLL-2003, AG, and SEC datasets in Table 3. The descriptions are from (Sang & De Meulder, 2003) and Spacy[3], supporting eighteen different entity types. In the current work, we play with four different types and their combinations. Note that, in UeDP, providing the name of an algorithm and a sensitive entity means we consider that type of entity as sensitive entities in the training process. For instance, in Figure 6, UeDP-Alg $f_{\mathcal{E}^+}$ (Org) means we use all organization entities as sensitive entities in the UeDP Optimized algorithm. "All entities" means all types of sensitive entities considered for the dataset are used. For example, "all entities" in the CONLL-2003 dataset means all person, location, organization, and miscellaneous entities are regarded as sensitive entities. Meanwhile, in the AG and SEC datasets, "all entities" means that all organization, location, GPE, and PII entities are considered sensitive entities. More entity types are also presented in Table 3 so that users can have more choices when identifying sensitive entities.

Table 3: Description of sensitive entity categories.

Type	Description
Person	Person, i.e., people, including fictional
Loc	Location, i.e., non-GPE locations, mountain ranges, bodies of water.
Org	Organization, i.e., companies, agencies, institutions, etc.
Misc	Miscellaneous, i.e., entities that do not belong to the person, location, and organization in CONLL-2003.
GPE	Geopolitical entity, i.e., countries, cities, states
PII	Personal identification information, i.e., person name, location, phone number, etc.
Date	Absolute or relative dates or periods
NoRP	Nationalities or religious or political groups
Fac	Buildings, airports, highways, bridges, etc.
Product	Objects, vehicles, foods, etc. (Not services.)
Event	Named hurricanes, battles, wars, sports events, etc.
Law	Named documents made into laws
Language	Any named language
Work of art	Titles of books, songs, etc.
Time	Times smaller than a day
Percent	Percentage, including "%"
Money	Monetary values, including unit
Quantity	Measurements, as of weight or distance
Ordinal	"First", "second", etc.
Cardinal	Numerals that do not fall under another type

[3] https://spacy.io/api/annotation#named-entities

C PSEUDO-CODE OF UEDP PRESERVING ALGORITHM

Algorithm 1 UeDP-Alg

Input: Dataset D, set of sensitive entities E, set of sensitive samples S, set of non-sensitive samples $\overline{S}$, user sampling rate q_u, sensitive entity sampling rate q_e, non-sensitive sample sampling rate $q_{\overline{s}}$, a hyper-parameter z, gradient clipping bound β, and number of iterations T

Initialize model θ^0 and moments accountant $\mathcal{M}$

$w_u = \min(\frac{n_u}{\hat{w}_u}, 1)$ for all users u (n_u is the number of samples in user u, $\hat{w}_u$ is per-user sample cap)

$w_e = \min(\frac{n_e}{\hat{w}_e}, 1)$ for all sensitive samples in S (n_e is the number of sensitive samples containing sensitive entities e, $\hat{w}_e$ is per-entity sample cap)

$w_s = \min(\frac{n_s}{\hat{w}_s}, 1)$ for all non-sensitive samples in $\overline{S}$ (n_s is the number of non-sensitive samples in $\overline{S}$, $\hat{w}_s$ is per-sample sample cap)

$W_u = \sum_u w_u$, $W_e = \sum_{e \in S} w_e$, $W_{\overline{s}} = \sum_{s \in \overline{S}} w_s$

for $t \in T$ **do**

  $U^t \leftarrow$ sample users with probability q_u

  for each user $u \in U^t$ **do**

    $S_u^t \leftarrow$ sensitive samples (belonging to the user u) consisting of sensitive entities E^t sampled from E with probability q_e

    $\begin{cases} \Delta_{u,\mathcal{E}}^{t+1} \leftarrow \textbf{Simple-Update}(u, S_u^t, \theta^t, \text{ClipFn}), \text{for } f_{\mathcal{E}} \text{in } \textbf{Lemma 1} \\ \Delta_{u,\mathcal{E}}^{t+1} \leftarrow \textbf{Optimized-Update}(u, S_u^t, \theta^t, \text{ClipFn}), \text{for } f_{\mathcal{E}^+} \text{ in } \textbf{Lemma 2} \end{cases}$

  $\Delta^{t+1} = \begin{cases} \frac{\sum_{u \in U^t} w_u \Delta_{u,\mathcal{E}}^{t+1}}{q_u W_u \times q_e W_e}, \text{for } f_{\mathcal{E}} \\ \frac{\sum_{u \in U^t} w_u \Delta_{u,\mathcal{E}}^{t+1}}{q_u W_u \times (q_e W_e + q_{\overline{e}} W_{\overline{e}})}, \text{for } f_{\mathcal{E}^+} \end{cases}$

  $\sigma \leftarrow \begin{cases} \frac{z(q_u|U|+1)\max(w_u)\beta}{q_u W_u \times q_e W_e}, \text{for } f_{\mathcal{E}} \\ \frac{z(q_u|U|+1)\max(w_u)\beta}{q_u W_u \times (q_e W_e + q_{\overline{e}} W_{\overline{e}})}, \text{for } f_{\mathcal{E}^+} \end{cases}$

  $\theta^{t+1} \leftarrow \theta^t + \Delta^{t+1} + \mathcal{N}(0, I\sigma^2)$

  $\mathcal{M}$.accum_priv_spending(z)

print $\mathcal{M}$.get_priv_spent()

Output: (ϵ, δ)-UeDP θ, $\mathcal{M}$

Simple-Update$(u, S_u^t, \theta^t, \text{ClipFn})$:

  for each sample s in S_u^t **do**

    $\theta \leftarrow \theta^t - \eta \bigtriangledown l(\theta, s)$

    $\Delta_{u,s} = \theta - \theta^t$

  $\Delta_{u,\mathcal{E}} = \sum_{e \in E^t} w_e (\sum_{s \in S_{ue}^t} \Delta_{u,s})$

  return ClipFn($\Delta_{u,\mathcal{E}}$, β)

Optimized-Update$(u, S_u^t, \theta^t, \text{ClipFn})$:

  $\overline{S}_u^t \leftarrow$ non-sensitive samples sampled from $\overline{S}_u$ with probability $q_{\overline{s}}$

  for each sample s in $S_u^t \cup \overline{S}_u^t$ **do**

    $\theta \leftarrow \theta^t - \eta \bigtriangledown l(\theta, s)$

    $\Delta_{u,s} = \theta - \theta^t$

  $\forall e \in E^t : \Delta_{u,e} = w_e (\sum_{s \in S_{ue}^t} \Delta_{u,s})$

  $\Delta_{u,\mathcal{E}} = (\sum_{e \in E^t} w_e \Delta_{u,e} + \sum_{s \in \overline{S}_u^t} w_s \Delta_{u,s})$

  return ClipFn($\Delta_{u,\mathcal{E}}$, β)

ClipFn(Δ, β):

  return $\pi(\Delta, \beta) = \Delta \cdot \min\left(1, \frac{\beta}{\|\Delta\|}\right)$

D PROOF OF LEMMA 1

Proof. If for all users u we have $\|\Delta_{u,\mathcal{E}}^{t+1}\|_2 \leq \beta$, then

$$\begin{aligned}\mathbb{S}(f_{\mathcal{E}}) &= \frac{\left[\sum_{u\in U^t\cup u'} w_u[(\sum_{e\in E^t} w_e(\sum_{s\in S_{ue}^t}\Delta_{u,s})) + w_{e'}(\sum_{s\in S_{ue'}^t}\Delta_{u,e'})]\right]}{(q_u W_u \times q_e W_e)} \\ &\quad - \frac{\sum_{u\in U^t} w_u[\sum_{e\in E^t} w_e(\sum_{s\in S_{ue}^t}\Delta_{u,s})]}{(q_u W_u \times q_e W_e)} \\ &\leq \frac{\sum_{u\in U^t\cup u'} w_u\beta}{q_u W_u \times q_e W_e} \leq \frac{(q_u|U|+1)\max(w_u)\beta}{q_u W_u \times q_e W_e} \end{aligned} \tag{8}$$

Consequently, Lemma 1 holds. □

E PROOF OF LEMMA 2

Proof. If for all users u we have $\|\Delta_{u,\mathcal{E}}^{t+1}\|_2 \leq \beta$, then

$$\begin{aligned}\mathbb{S}(f_{\mathcal{E}^+}) &= \frac{\left[\sum_{u\in U^t\cup u'} w_u[(\sum_{e\in E^t} w_e(\sum_{s\in S_{ue}^t}\Delta_{u,s})) + w_{e'}(\sum_{s\in S_{ue'}^t}\Delta_{u,e'}) + \sum_{s\in\overline{S}_u^t} w_s\Delta_{u,s}]\right]}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})} \\ &\quad - \frac{\sum_{u\in U^t} w_u[\sum_{e\in E^t} w_e(\sum_{s\in S_{ue}^t}\Delta_{u,s}) + \sum_{s\in\overline{S}_u^t} w_s\Delta_{u,s}]}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})} \\ &\leq \frac{\sum_{u\in U^t\cup u'}[(w_u)\beta]}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})} \leq \frac{(q_u|U|+1)\max(w_u)\beta}{q_u W_u \times (q_e W_e + q_{\overline{s}} W_{\overline{s}})} \end{aligned} \tag{9}$$

Consequently, Lemma 2 holds. □

F PROOF OF THEOREM 1

Proof. At each step, users, sensitive entities, and non-sensitive samples are respectively selected randomly with probability q_u, with probability q_e, and with probability $q_{\overline{s}}$. For the estimator $f_{\mathcal{E}^+}$, if the l_2-norm of each user's gradient update, using both sampled sensitive and non-sensitive samples, is bounded by $\mathbb{S}(f_{\mathcal{E}^+})$, then the moments can be bounded by that of the sampled Gaussian mechanism with sensitivity 1, the scale $z = \sigma/\mathbb{S}(f_{\mathcal{E}^+})$, and sampling probabilities q_u, q_e, and $q_{\overline{s}}$. Thus, we can apply the composability as in Theorem 2.1 (Abadi et al., 2016) to correctly compute the UeDP privacy loss with the scale $z = \sigma/\mathbb{S}(f_{\mathcal{E}^+})$ for T steps.

Similarly, for $f_{\mathcal{E}}$, we can use the composability as in Theorem 2.1 (Abadi et al., 2016) to compute the UeDP privacy loss with the scale $z = \sigma/\mathbb{S}(f_{\mathcal{E}})$ for T training steps. □

G REVISITING WORD-LEVEL LDP ANALYSIS IN (LYU ET AL., 2020A)

This section aims at revisiting privacy protection in (Lyu et al., 2020a) and describes existing issues of privacy accumulation over the embedding dimension in terms of theory and experimental results of their approach, which is confirmed by the authors (Lyu et al.). Then, we provide corrected Theorems based upon our theoretical analysis, and we compare them with our approaches.

In the paper, they aim at preserving privacy of the extracted test representation from the user while maintaining a good performance of the classifier, which is trained at a server by the data collected from users. To achieve the goal, they consider a word-level DP; i.e., two inputs x and x' are considered to be adjacent if they differ by at most 1 word. Additionally, they introduce a DP noise layer r after a predefined feature extractor $f(x)$. To train a robust classifier at the server, they add the same level of noise as the test phase in the training process and optimize the classifier by minimizing the loss function as follows:

$$\mathcal{L}(x, y) = \mathcal{X}(C(f(x) + r), y) \tag{10}$$

where C is the classifier, y is the true label, $\mathcal{X}$ is the cross entropy loss function. The Laplace noise layer r is injected into the embedding $f(x)$ in which its coordinates $r = \{r_1, r_2, \ldots, r_k\}$ are i.i.d. random variables drawn from the Laplace distribution defined by $Lap(b)$ with $b = \frac{\Delta_f}{\epsilon}$, ϵ is the privacy budget, and Δ_f is the sensitivity of the extracted representation. Here, k is the dimension of the embedding $f(x)$.

Algorithm 2 describes how to derive differentially private representation from the feature extractor f. Note that x_s in the Algorithm 2 is a sentence (equivalent to x in our notation), which is considered to be sensitive and be protected.

Algorithm 2 Differentially Private Neural Representation (DPNR) (Lyu et al., 2020a)

1: **Input**: Each sensitive input $x_s \in \mathbb{R}^d$, feature extractor f
2: **Parameters**: Dropout vector $I_n \in \{0, 1\}^d$
3: Word dropout: $\tilde{x}_s \leftarrow x_s \odot I_n$, where $\odot$ performs a word-wise multiplication.
4: Extraction: $x_r \leftarrow f(\tilde{x}_s)$
5: Normalization: $x_r \leftarrow x_r - \min(x_r)/(\max(x_r) - \min(x_r))$
6: Perturbation: $\hat{x_r} \leftarrow x_r + r$, $r_i \sim Lap(b)$
7: **Output**: Perturbed representation $\hat{x_r}$.

Theorems 1 and 2 in (Lyu et al., 2020a) do NOT hold. Lyu et al. (2020a) consider adjacent databases differing by one word. It is clear that changing one word in x may result in changing the entire embedding vector $f(x)$. In their paper, they normalize each element of $f(x)$ into the range $[0, 1]$ (Line 5, Algorithm 2), hence each element sensitivity of $f(x)$ is $\Delta_f = 1$, the noise is $Lap(\Delta_f/\epsilon)$. Therefore, each element of the embedding $f(x)$ consumes a privacy budget ϵ. Since the k elements of the embedding are derived from a single sensitive input x, applying the LDP mechanism $\mathcal{A}(.)$, i.e., $Lap(b)$, k times will consume the privacy budget $k\epsilon$. This follows the composition property in DP, also known as the curse of dimensionality in local DP. Note that the k elements cannot be treated by using the parallel property in DP (Dwork & Lei, 2009), since all of them are derived from a single input x (data), NOT from k different inputs (k different datasets). Consequently, Theorem 1 of (Lyu et al., 2020a) does NOT hold at their reported ϵ word-level DP. Since Theorem 2 of (Lyu et al., 2020a) is derived from the result of Theorem 1 of (Lyu et al., 2020a), it is clear that Theorem 2 of (Lyu et al., 2020a) also does NOT hold.

Element-level DP does NOT hold. During our discussion with the authors of (Lyu et al., 2020a), the authors mentioned that their approach preserves a new notion of $(\epsilon, 0)$-element-level DP, i.e., two embeddings differ from one element, instead of a word-level DP. However, for the element-DP to hold, all the elements in the embedding $f(x)$ must be independent from each other, that is, changing one element will not result in changing any other element. If changing one element results in changing all the remaining elements, then element-DP will be suffered from the dimension of the embedding by following group privacy. In the current approach, changing one element means there is a change in the input data x to occur. Equivalently, using BERT, any change in the input data x will result in changing the whole embedding (all elements). Therefore, the condition of two neighboring embeddings only differing in only one element does NOT hold in theory and practice. Consequently, the introduced element-level DP does NOT hold at the level of $(\epsilon, 0)$-DP.

Surprisingly Good Experimental Results in (Lyu et al., 2020a) due to Inappropriate Privacy Analysis. In their experimental results, e.g., Table 2 of (Lyu et al., 2020a), it is surprising that the approach could achieve almost the same (and even better) model utility with noiseless model given the extremely low $\epsilon = 0.05$ using BERT embeddings. For Theorems 1 and 2 to hold and support the correctness of the approach, the privacy budget must be $\epsilon \times k$, which is at least $0.05 \times 768 = 38.4$, instead of just ϵ reported in the paper. Similar results were reported through out the all in experiments. Eventually, the approach in (Lyu et al., 2020a) cannot provide any practical DP protection to the embedding at any levels, including word-level, character-level, and even a single embedding element given an input data x.

Our Correcting Theorems 1 and 2 in (Lyu et al., 2020a). Based upon our analysis, we introduce corrected versions of the Theorems 1 and 2 in (Lyu et al., 2020a), as follows.

Theorem 1. ***Corrected Theorem 1 in (Lyu et al., 2020a).*** *Let the entries of the noise vector r be drawn from $Lap(b)$ with $b = \frac{\Delta_f}{\epsilon}$. The Algorithm 2 is $k\epsilon$-word-level DP, where k is dimension of the embedding $f(x)$.*

Proof. Each element of the embedding f is bounded in $[0, 1]$, so $\Delta_f = 1$ for each element. By adding i.i.d. random noise variables drawn from the Laplace $Lap(b)$ with $b = \frac{\Delta_f}{\epsilon}$ into each element of f, each element consumes ϵ/k privacy budget. Since the k elements of the embedding are derived from a single sensitive input x, applying the mechanism $Lap(b)$ k times on the k elements will consume the privacy budget $k\epsilon$. Therefore, the Algorithm 2 is $k\epsilon$-word-level DP. □

Theorem 2. *Given an input $x \in D$, suppose $\mathcal{A}(x) = f(x) + r$ is $k\epsilon$-word-level DP, let I_n with dropout rate μ be applied to x: $\tilde{x} = x \odot I_n$, then $\mathcal{A}(\tilde{x})$ is ϵ'-word level-DP, where $\epsilon' = \ln[(1 - \mu)\exp(k\epsilon) + \mu]$.*

Proof. Suppose there are two adjacent inputs x_1 and x_2 that differ only in the i-th coordinate (word), say $x_{1i} = v$, $x_{2i} \neq v$. For arbitrary binary vector I_n, after dropout, $\tilde{x}_1 = x_1 \odot I_n$, $\tilde{x}_2 = x_2 \odot I_n$, there are two possible cases, i.e., $I_{ni} = 0$ and $I_{ni} = 1$.

If $I_{ni} = 0$: Since x_1 and x_2 differ only in i-th coordinate, after dropout $\tilde{x}_{1i} = \tilde{x}_{2i} = 0$, hence $x_1 \odot I_n = x_2 \odot I_n$. Then $Pr\{\mathcal{A}(x_1 \odot I_n) = S\} = Pr\{\mathcal{A}(x_2 \odot I_n) = S\}$.

If $I_{ni} = 1$: Since x_1 and x_2 differ only in i-th coordinate, after dropout $\tilde{x}_{1i} = v$, and $\tilde{x}_{2i} \neq v$. Since $\mathcal{A}(x)$ is $k\epsilon$-word level-DP, then $Pr\{\mathcal{A}(x_1 \odot I_n) = S\} \leq \exp(k\epsilon) Pr\{\mathcal{A}(x_2 \odot I_n) = S\}$.

Combining these two cases, and $Pr[I_{ni} = 0] = \mu$, we have:

$$\begin{aligned} Pr\{\mathcal{A}(x_1 \odot I_n) = S\} &= \mu Pr\{\mathcal{A}(x_1 \odot I_n) = S\} + (1 - \mu) Pr\{\mathcal{A}(x_1 \odot I_n) = S\} \\ &\leq \mu Pr\{\mathcal{A}(x_2 \odot I_n) = S\} + (1 - \mu)\exp(k\epsilon) Pr\{\mathcal{A}(x_2 \odot I_n) = S\} \\ &= [(1 - \mu)\exp(k\epsilon) + \mu] Pr\{\mathcal{A}(x_2 \odot I_n) = S\} \\ &= \exp\Big(\ln[(1 - \mu)\exp(k\epsilon) + \mu]\Big) Pr\{\mathcal{A}(x_2 \odot I_n) = S\} \end{aligned} \tag{11}$$

Therefore, after dropout, the privacy budget is $\epsilon' = \ln[(1 - \mu)\exp(k\epsilon) + \mu]$. □

Comparison with Our Work. Apart from the privacy accumulation over the embedding dimension issue, in their work, during training the model, they draw the Laplace or Gaussian noise at every training iteration. It means that the model accesses the raw data at every iteration; therefore, the privacy budget at the training phase is accumulated over the number of training iterations, which can be a large number that results in an exploded privacy budget in training. In fact, they focus on protecting privacy at the inference time and use the noise in the training phase to obtain a more robust model without considering training data privacy. This is different from our goal to protect users and sensitive entities of training data, which is a more challenging task. Our UeDP-preserving model can be deployed to the end-users for a direct use in the inference phase, without demanding that the end-users send their data embedding to our server; thus offering a more rigorous privacy protection and better usability. In addition to this, our approach offer a tight DP budget bound compared with the DPNR algorithm in (Lyu et al., 2020a), which consumes large DP budgets that is proportional to the large size of the embedding k.

H Revisiting Binary Encoding-based LDP Analysis in (Lyu et al., 2020b)

This section aims at revisiting privacy protection for the binary encoding-based LDP mechanism in (Lyu et al., 2020a) and it describes the existing privacy exaggeration problem and the loose privacy budget bounding problem in terms of theoretical results of their approach.

In general, let us apply a randomized response mechanism $B(v)$ on a binary bit string v to preserve LDP. We denote $B_i(v)$ when $B(v)$ is applied on bit i-th of v. If $B(v)$ is ϵ-LDP applied on all elements of the embedding v, then the privacy budget ϵ consumed by v is computed from:

$$\frac{P_{B(v)}(z)}{P_{B(v')}(z)} = \frac{\exp(-\frac{\epsilon d(B(v)-z)}{\Delta B})}{\exp(-\frac{\epsilon d(B(v')-z)}{\Delta B})} \leq \exp(\frac{\epsilon|d(B(v)-z)-d(B(v')-z)|}{\Delta B}) \leq \exp(\frac{\epsilon d(B(v)-B(v'))}{\Delta B}) \tag{12}$$

where $\Delta B = \max_{v,v'} |B(v) - B(v')|$ is the sensitivity of the function $B(v)$.

H.1 Hash Function-based LDP (Erlingsson et al., 2014; Bassily & Smith, 2015; Wang et al., 2017)

If v is obtained by mapping a value or an input data into a random hash using a hash function, then all elements of v are i.i.d., and changing one bit of v will not result in changing any other bits of v. Therefore, $\Delta B = \sum_i \Delta B_i$, in which ΔB_i is the sensitivity of $B_i(v)$ computed by $\Delta B_i = \max_{v,v'} |B_i(v) - B_i(v')| = \max_{v,v'} \sum_{j=1}^{rl} |B_i(v_j) - B_i(v'_j)| = 1$, and $d(B_i(v) - B_i(v')) = 1$ since v_i and v'_i are adjacent databases.

Therefore, in hash function-based approaches (Erlingsson et al., 2014; Bassily & Smith, 2015; Wang et al., 2017), we have:

$$\frac{P_{B(v)}(z)}{P_{B(v')}(z)} = \Pi_{i=1}^{rl} \frac{P_{B_i(v)}(z)}{P_{B_i(v')}(z)} \leq \Pi_{i=1}^{rl} \exp(\frac{\epsilon_i d(B_i(v) - B_i(v'))}{\Delta B_i}) \leq \Pi_{i=1}^{rl} \exp(\epsilon_i) = \exp(\sum_i \epsilon_i) \tag{13}$$

As a result, the privacy budget spent on v is truthfully ϵ where $\epsilon = \sum_i \epsilon_i$.

H.2 Binary Encoding-based LDP (Lyu et al., 2020b)

As discussed in Section 2, binary encodings of all the embedded features are concatenated into a large binary vector v of the size rl, where r is the number of embedded features, and l is the number of binary bits, i.e., 1 sign bit, m integer bits, and n fraction bits, used to encode each embedded feature's real value, i.e., $l = 10$ in (Lyu et al., 2020b). Straightforwardly, Lyu et al. (2020b) applied the randomized response mechanism B on the bit string rl and shown that they can achieve ϵ-LDP:

$$\frac{P_{B(v)}(z)}{P_{B(v')}(z)} = \Pi_{i=1}^{rl} \frac{P_{B_i(v)}(z)}{P_{B_i(v')}(z)} \leq \exp(\epsilon) \tag{14}$$

However, a critical mistake in their paper is considering $\Delta B = rl$ for the entire embedding v of size rl, which is equivalent to consider $\Delta B_i = 1$ for every bit. In addition, a binary encoding bit string cannot be treated as a random hash, since the released result of $B_i(v)$ can be transformed into a real-value exposing privacy risk to the true value v. This is fundamentally different for a hash, in which the perturbation cannot be transformed into a real-value. In other words, the bits in binary encoding is not independent as binary bits in a hash.

From Eq. 14, for every bit i, there always exists a privacy budget ϵ_i quantified by $\frac{P_{B_i(v)}(z)}{P_{B_i(v')}(z)}$. Note that $\epsilon = \sum_i \epsilon_i$. This is equivalent to satisfying the following condition: $\frac{P_{B_i(v)}(z)}{P_{B_i(v')}(z)} \leq \exp(\frac{\epsilon_i d(B_i(v)-B_i(v'))}{\Delta B_i}) \leq \exp(\epsilon_i)$. Next, we will show that this condition does NOT always hold given the binary encoded bit string rl.

Given $\Delta B_i = 1$ for every bit, then we have:

$$\frac{P_{B_i(v)}(z)}{P_{B_i(v')}(z)} \leq \exp(\frac{\epsilon_i d(B_i(v) - B_i(v'))}{\Delta B_i}) = \exp(\epsilon_i d(B_i(v) - B_i(v')))$$

To bound the privacy loss given the $B_i(v)$, we need to bound the distance function $d(B_i(v)-B_i(v'))$.

Privacy Risk Exaggeration. Given a bit i in the integer part of one embedding element, we have that:

$$\text{If } i \in [1, m] \text{ is an integer bit : } d(B_i(v) - B_i(v')) \leq \max_{v,v'}(|B_i(v) - B_i(v')| \times 2^{i-1}) \tag{15}$$

We can see that $|B_i(v) - B_i(v')| \leq 1$. As a result, we have

$$\text{If } i \in [1, m] \text{ is an integer bit : } d(B_i(v) - B_i(v')) \leq 2^{i-1} \tag{16}$$

Consequently, we have that

$$\text{If } i \in [1, m] \text{ is an integer bit : } \frac{P_{B_i(v)}(z)}{P_{B_i(v')}(z)} \leq \exp(\frac{\epsilon_i d(B_i(v) - B_i(v'))}{\Delta B_i}) \leq \exp(2^{i-1}\epsilon_i) \tag{17}$$

From the Eq. 18, the actual privacy budget used for an integer bit $i \in [1, m]$ is $2^{i-1}\epsilon_i$, which is significantly larger than the privacy budget ϵ_i quantified by the binary encoding-based LDP approach in (Lyu et al., 2020b). We call this a privacy risk exaggeration in the integer part of the binary encoding-based LDP approach in (Lyu et al., 2020b). This problem is also true for the sign bit. If i is a sign bit, then the actual privacy budget is $2^{m+1}\epsilon_i$, since $d(B_i(v) - B_i(v')) \leq 2^{m+1}$, as follows:

$$\text{If } i \text{ is a sign bit : } \frac{P_{B_i(v)}(z)}{P_{B_i(v')}(z)} \leq \exp(\frac{\epsilon_i d(B_i(v) - B_i(v'))}{\Delta B_i}) \leq \exp(2^{m+1}\epsilon_i) \tag{18}$$

This privacy risk exaggeration problem is severe, since the actual privacy budget can be exponential, given $m \times r$ integer bits and r sign bits in the whole bit string rl.

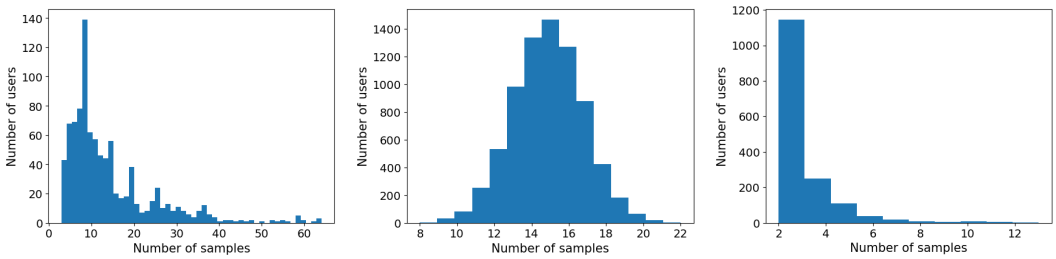
Loose Privacy Budget Bounding. Similarly, for fraction bits $i \in [1, n]$, we have that

$$\text{If } i \in [1, n] \text{ is a fraction bit : } \frac{P_{B_i(v)}(z)}{P_{B_i(v')}(z)} \leq \exp(\frac{\epsilon_i d(B_i(v) - B_i(v'))}{\Delta B_i}) \leq \exp(2^{-i}\epsilon_i) \tag{19}$$

From Eq. 19, the actual privacy budget used for a fraction bit $i \in [1, n]$ is $2^{-i}\epsilon_i$, which is smaller than the privacy budget ϵ_i quantified in the binary encoding-based LDP approach in (Lyu et al., 2020b). In other words, the approach proposed in (Lyu et al., 2020b) quantified the privacy budget more than it needed. We call this a loose privacy budget bounding problem in the fraction part in (Lyu et al., 2020b).

To conclude, straightforwardly applying a randomized response mechanism on binary encoded vector as in (Lyu et al., 2020b) cannot correctly bound the actual privacy risk with local DP, due to the primitive difference between a random hash and a binary encoding bit string. The privacy risk exaggeration problem can severely loosen the privacy protection claimed in (Lyu et al., 2020b). Similar approaches with (Lyu et al., 2020b), such as (Chamikara et al., 2019), may suffer from the same problems.

I Datasets and Data Processing



(a) CONLL-2003 dataset (b) AG dataset (c) SEC dataset

Figure 5: Distribution of users and samples in CONLL-2003, AG, and SEC datasets.

CONLL-2003 consists of Reuters news stories published between August 1996 and August 1997. CONLL-2003 is an NER dataset, where there are labels for four different types of named entities, including location, organization, person, and miscellaneous entities. These types of named entities are considered sensitive entities. In the CONLL-2003 dataset, there is no obvious user information; hence, we consider each document as a user and each sentence in a document as a sample in the next word prediction task.

AG dataset is a collection of news articles gathered from more than $2,000$ news sources by Cometomyhead academic news search engine. It is categorized into four classes: world, sport, business, and science/technology classes. Similar to the CONLL-2003 dataset, there is no user information in the AG dataset. To imitate a user indicator, we randomly divide news into different users based on Gaussian distribution. There are no named entities; thus, we apply pre-trained Spacy to find named entities and PII in the dataset. We choose different types of these named entities to be sensitive entities: organization, GPE (i.e., countries, cities, and states), location, and PII entities. Please refer to Table 3 in **Appendix B** for details of sensitive entities.

Our SEC dataset consists of over 10,000 contract clauses collected from contracts submitted in SEC filings[4]. Since the contracts can be associated with a company ID, we use the ID as a user indicator. Similar to the AG dataset, we consider organization, GPE, location, and PII entities as sensitive entities to protect.

In addition to the next word prediction in all datasets, we conducted text classification on the AG dataset to further strengthen our observations. For text classification, the number of labels is not sufficient in the SEC dataset, and the labels do not exist in the CONLL-2003 dataset. Therefore, we do not utilize CONLL-2003 and SEC datasets for text classification in this study.

For data preprocessing, we changed all words to lower-case and removed punctuation marks. In CONLL-2003 and SEC datasets, rare words that appear less than three times in the dataset are replaced with a $<$*unk*$>$ token. In the AG dataset, we kept $30,000$ words that appear the most, and we replaced the rest with a $<$*unk*$>$ token.

Figure 5 shows the distribution of users and samples in CONLL-2003, AG, and SEC datasets. In the CONLL-2003 dataset, there is no obvious user information; hence, we consider each document as a user and each sentence in a document as a sample. Like the CONLL-2003 dataset, in the AG dataset, there is no user information. Therefore, to imitate a user indicator, we randomly divide news into different users. The number of samples per user follows a Gaussian distribution $\mathcal{N}(15, 2^2)$, i.e., there are 15 samples per user on average, and the standard deviation is 2 samples. Each news is considered as a sample. In the SEC dataset, since the contracts can be associated with a company ID, we use the ID as a user indicator. The document related to the ID is considered as a sample.

J Model Configuration

For the GPT-2 and BERT models, we use the versions that have 12-layer, 768-hidden, and 12-heads. Adam optimizer is used with the learning rate is 10^{-5}. Clipping bound $\beta = 0.1$ and the scale $z = 2.5$. The sampling rate q_u, q_e, $q_{\overline{s}}$ are 0.05, 0.5, and 1.0.

With the AWD-LSTM model for both next word prediction and text classification tasks, all experiments use a three-layer LSTM model with 1,150 units in the hidden layer and an embedding input layer of size 100. Embedding weights are uniformly initialized in the interval $[-0.1, 0.1]$ with dimension $d = 100$ and other weights are initialized between $[-\frac{1}{\sqrt{H}}, \frac{1}{\sqrt{H}}]$, where H is the size of all hidden layers. The values used for dropout on the embedding layer, the LSTM hidden-to-hidden matrix, and the final LSTM layer's output are 0.1, 0.3, 0.5, respectively. Clipping bound $\beta = 0.1$ and the scale $z = 2$. The sampling rate q_u, q_e, $q_{\overline{s}}$ are 0.05, 0.5, and 1.0 (note that $q_{\overline{s}}$ is 0.6 in the text classification task). SGD optimizer is used. In text classification with the AG dataset, a softmax layer is applied on top of the AWD-LSTM with the output dimension is 4, corresponding to four classes in the dataset. The same sets of sensitive entity categories are used in the next word prediction and the text classification tasks.

[4] https://www.sec.gov/edgar.shtml

K EXPERIMENTAL RESULTS ON DIFFERENT CATEGORIES OF SENSITIVE ENTITIES AND TEXT CLASSIFICATION

From theoretical analysis, as shown in Figure 6, the greater the number of sensitive data samples to protect, the higher the privacy budget is needed, and the lower the performance that the language model achieves (i.e., the model reaches higher values of perplexity). These theoretical and experimental results are consistent with our theoretical analysis after Lemma 2.

For instance, in the SEC dataset, the number of sensitive samples in each category is 60 in GPE, 273 in location, 357 in PII, $1,955$ in organization, and $2,166$ in all entities. After 500 steps, the respective values of ϵ are 0.19 in GPE, 0.24 in location, 0.26 in PII, 0.73 in organization, 0.81 in all entities, and 4.08 in User-level DP. In addition, at $\epsilon = 0.5$, we obtain perplexity values of 39.41 in GPE, 58.11 in location, 76.05 in PII, 235.32 in organization, 277.42 in a 556.34 in User-level DP.

Given a little larger privacy budget $\epsilon \geq 2$, the perplexity values drop, and the gap among different sensitive entity categories, User-level DP, and the noiseless AWD-LSTM model reduces notably. For instance, at $\epsilon = 3$, the perplexity value is 34.79 in organization, and 36.25 in all entities, compared with 45.18 in User-level DP, and 32.77 in the noiseless AWD-LSTM model.

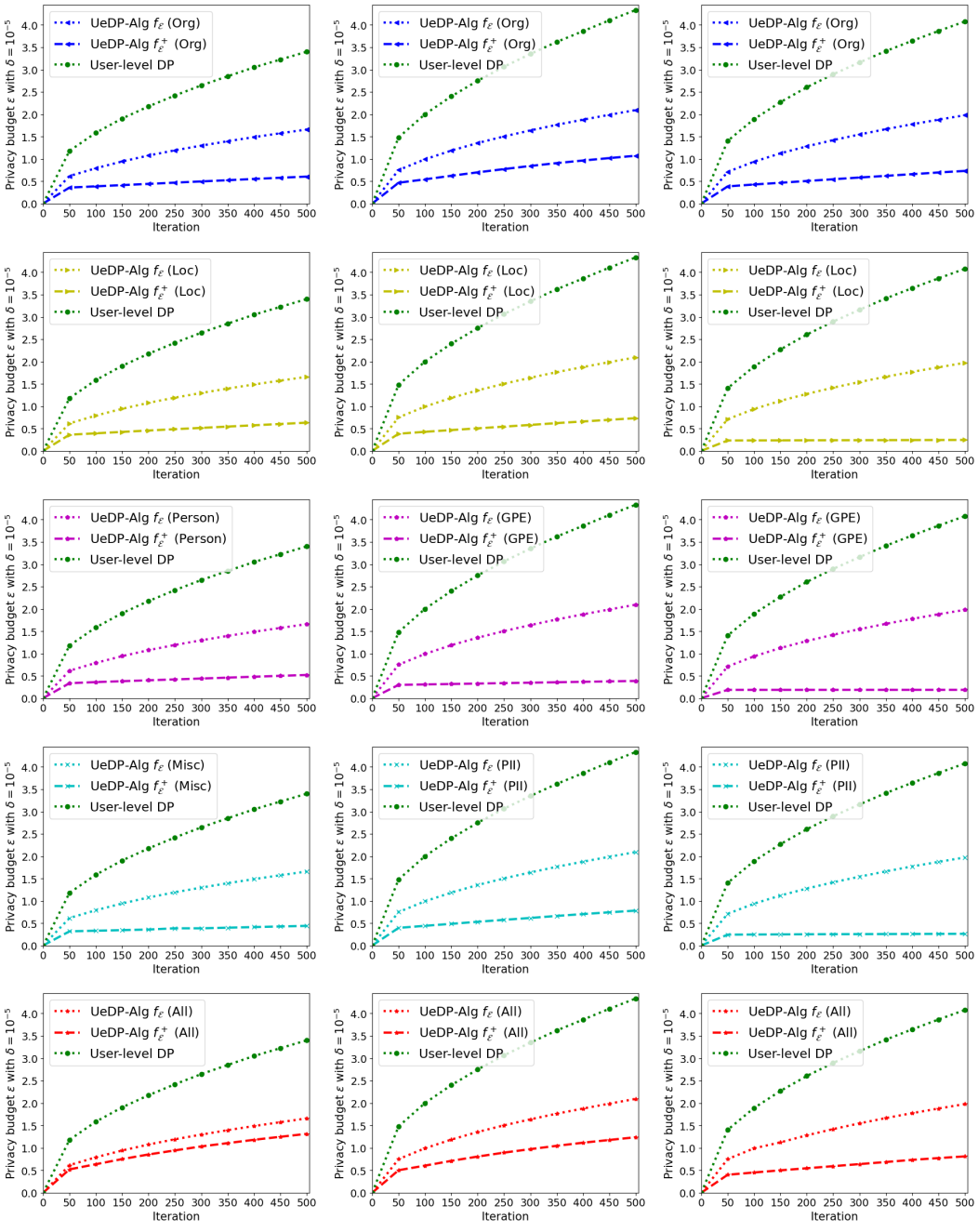
In the CONLL-2003 dataset, the number of sensitive samples per category is $3,438$ in miscellaneous, $4,406$ in person, $5,187$ in organization, $5,433$ in location, and $11,176$ in all entities. There are $14,040$ samples in total. At iteration 500th, the corresponding ϵs are 0.44 in miscellaneous, 0.52 in person, 0.61 in organization, 0.63 in location, and 1.31 in all entities, and 3.40 in User-level DP. In the AG dataset, we find $18,506$ in GPE, $39,988$ in location, $42,683$ in PII, $58,177$ in organization, and $67,157$ in all entities. There are $112,000$ samples in total. At iteration 500th, the corresponding ϵs are: 0.39 in GPE, 0.73 in location, 0.78 in PII, 1.07 in organization, and 1.24 in all entities, and 4.34 in User-level DP.

We can see that De-Identification achieves a competitive perplexity result. The key reason is that, in De-Identification, sensitive entities are marked, resulting in a smaller model sensitivity, compared with the worst-case scenarios (i.e., the upper bound of the model sensitivity) in our UeDP-Alg. Our algorithm offers rigorous DP guarantees for both users and sensitive entities; meanwhile, De-Identification provides no privacy guarantee to users or sensitive entities. More importantly, when we have a little larger privacy budget $\epsilon \geq 2$, our UeDP-Alg has very competitive perplexity values – even better than the De-Identification in all cases – approaching the noiseless AWD-LSTM model.

Like the next word prediction task, the text classification results on the AG dataset (Figure 10a) also showed that our UeDP-Alg achieves lower test error rates than baseline approaches in most cases across different types of sensitive entities. The result is promising given the very tight UeDP protection ($\epsilon \in [0.7, 1]$). For instance, at $\epsilon = 0.7$, the test error rates are 0.30 in GPE and in organization, 0.31 in location, 0.32 in PII, and 0.28 in all entities, compared with 0.32 in User-level DP and 0.32 in De-Identification.

When spending more privacy budget, the test error rates of both UeDP-Alg and User-level DP drops, approaching the noiseless AWD-LSTM model's upper-bound result. At $\epsilon = 1$, UeDP-Alg obtains 0.22 in organization, 0.26 in GPE, 0.28 in location, 0.27 in PII and in all entities, compared with 0.29 in User-level DP, 0.32 in De-Identification, and 0.22 in the noiseless AWD-LSTM model.

From our experiments, as shown in Figure 8, considering non-sensitive sentences into training UeDP notably improves model performance under the same UeDP protection. However, different tasks on different datasets may require different optimal values of the sampling rate $q_{\bar{s}}$ to achieve the best model performances. For instance, in the CONLL-2003 dataset, without non-sensitive sentences ($q_{\bar{s}} = 0$), the perplexity (PPL) is 27.06 in Organization, 30.67 in Person, 42.75 in Miscellaneous, 66.74 in Location, and 28.89 in All entities. At the same privacy budget, the lowest perplexity achieves at $q_{\bar{s}} = 0.4$ in Location (PPL = 21.85) and in Organization (PPL = 21.55), and at $q_{\bar{s}} = 0.6$ in Person (PPL = 20.10), in Miscellaneous (PPL = 19.56), and in All entities (PPL = 22.16). A similar phenomenon appears in the AG and SEC datasets. In the AG dataset, the PPL is significantly high without using non-sensitive sentences ($q_{\bar{s}} = 0$), such as PPL $= 4,058$ in GPE. When using non-sensitive sentences in training UeDP, the PPL significantly drops (i.e., the lower, the better), the lowest perplexity is achieved at $q_{\bar{s}} = 0.6$ in Location (PPL = 46.80) and in Organization (PPL = 54.30), in PII (PPL = 48.25), and in All entities (PPL = 59.62) and at $q_{\bar{s}} = 0.4$ in GPE (PPL = 45.97). In SEC dataset, the PPL is significantly high without using non-sensitive sentences ($q_{\bar{s}} = 0$), such



(m) CONLL-2003-all entities (n) AG-all entities (o) SEC-all entities

Figure 6: Privacy budget of UeDP-Alg $f_{\mathcal{E}}$, UeDP-Alg $f_{\mathcal{E}+}$, and User-level DP as a function of iterations in CONLL-2003, AG, and SEC datasets. UeDP-Alg $f_{\mathcal{E}+}$ achieves a tighter privacy budget compared with UeDP-Alg $f_{\mathcal{E}}$ and User-level DP.

as PPL $= 515.97$ in All entities. When using non-sensitive sentences in training UeDP, the PPL significantly drops (i.e., the lower, the better): the lowest perplexity is achieved at $q_{\bar{s}} = 0.6$ in Location (PPL = 36.47), at $q_{\bar{s}} = 0.8$ in Organization (PPL = 43.55), in GPE (PPL = 34.53), in PII (PPL = 33.78), and at $q_{\bar{s}} = 1.0$ (using all non-sensitive sentences in training) with All entities (PPL = 43.28).

• **Privacy Budget (ϵ, δ)-UeDP and Model Utility.** From our theoretical analysis, $f_{\mathcal{E}^+}$ is better than the estimator $f_{\mathcal{E}}$. Therefore, for the sake of simplicity, we only consider UeDP-Alg $f_{\mathcal{E}^+}$ instead of showing results from both estimators. From now, UeDP-Alg is used to indicate the use of our estimator $f_{\mathcal{E}^+}$. Figure 9 illustrates the perplexity as a function of the privacy budget ϵ for an AWD-LSTM model trained on a variety of sensitive entity categories in UeDP, User-level DP, and De-Identification. The noiseless AWD-LSTM model is considered an upper-bound performance mechanism without offering any privacy protection.

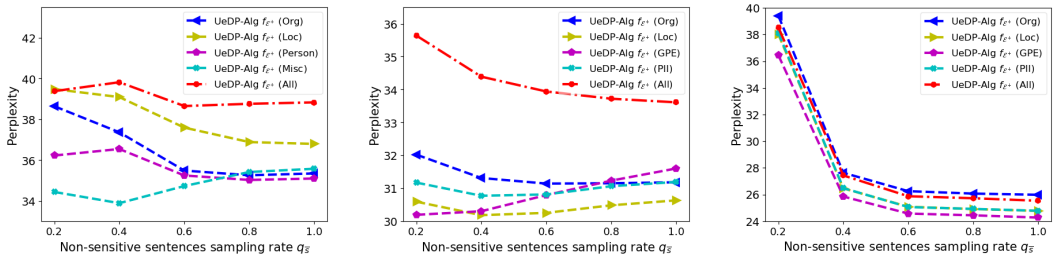
In the CONLL-2003 dataset (Figure 9a), there are NER labels for person, location, organization, and miscellaneous entities; therefore, we choose these types as sensitive entity categories to protect in UeDP-Alg. In all values of ϵ, UeDP-Alg achieves a better perplexity compared with User-level DP. Also, from $\epsilon = 1$ (reasonable privacy protection), our UeDP-Alg achieves a better perplexity than De-Identification. In fact, at $\epsilon = 1$, our UeDP-Alg achieves 25.76 for person, 25.09 for miscellaneous, 26.43 for organization, 26.45 for location entities, compared with 42.66 in User-level DP. When spending more privacy budget ($\epsilon \geq 1.5$), both UeDP-Alg and User-level DP converge at a very competitive perplexity level, approaching the upper-bound noiseless AWD-LSTM model. For instance, at $\epsilon = 2$, there are significant perplexity drops given UeDP-Alg and User-level DP mechanisms, i.e., our UeDP-Alg is 23.96 for person, 24.60 for organization, 24.88 for location, and 23.97 for miscellaneous entities. Meanwhile, the perplexity values of User-level DP, De-Identification, and the noiseless AWD-LSTM model are 26.18, 33.10, and 22.80.

Results on AG and SEC datasets (Figures 9b and 9c) further strengthen our observations. In AG and SEC datasets, we applied Spacy to identify different sensitive entity categories, such as GPE, location, organization, and PII (i.e., person and location information). UeDP-Alg achieves better results compared with User-level DP in all considering sensitive entity categories and privacy budgets, and outperforms De-Identification in most cases. That is promising and consistent with our previous analysis. For instance, in the AG dataset, at $\epsilon = 1$, our UeDP-Alg achieves 50.80 for PII, 50.90 for location, 51.58 for GPE, 52.42 for organization entities, compared with 55.96 in User-level DP. De-Identification obtains 58.57, and the upper bound result in the noiseless AWD-LSTM model is 47.93. Similarly, in the SEC dataset (Figure 9c), at $\epsilon = 1$, UeDP-Alg achieves perplexity of 33.39 in location, 40.07 in PII, 73.39 in organization, and 85.95 in all entities, compared with 99.36 in User-level DP, and 40.34 in De-Identification. In AG and SEC datasets, when we have a little larger privacy budget, i.e., $\epsilon \geq 0.9$ and $\epsilon \geq 2$ in AG and SEC datasets, our UeDP-Alg has better perplexity values than the De-Identification, approaching the noiseless AWD-LSTM model.

• **Sensitive Entity Categories.** In all datasets (Figures 6 and 9), the more sensitive data samples to protect, the higher the privacy budget is needed and the lower performance of the model achieves (i.e., higher values of perplexity). These theoretical and experimental results are consistent with our theoretical analysis after Lemma 2. For instance, in the SEC dataset, the number of sensitive samples in each category is as follows: 60 in GPE, 273 in location, 357 in PII, $1,955$ in organization, and $2,166$ in all entities. After 500 steps, the respective values of ϵ are 0.19 in GPE, 0.24 in location, 0.26 in PII, 0.73 in organization, 0.81 in all entities, and 4.08 in User-level DP (Figure 6). At $\epsilon = 0.5$ (Figure 9c), we obtain perplexity values of 39.41 in GPE, 58.11 in location, 76.05 in PII, 235.32 in organization, 277.42 in all entities, and a 556.34 in User-level DP.

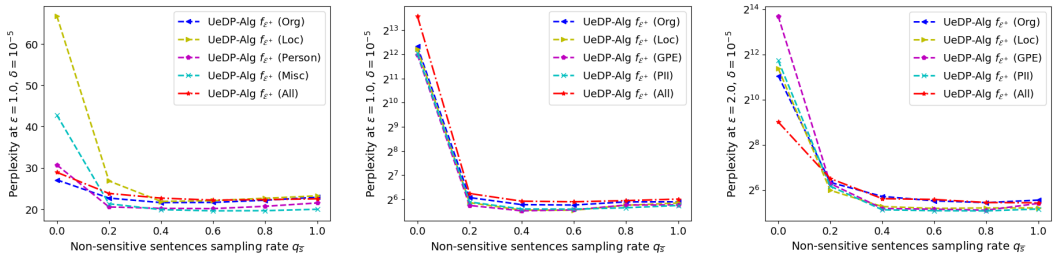
• **Text classification.** Figure 10a shows that our UeDP-Alg achieves lower test error rates in terms of text classification on the AG dataset than baseline approaches in most cases across different types of sensitive entities under a very tight UeDP protection ($\epsilon \in [0.7, 1]$). This is a promising result. When ϵ is higher, the test error rates of both UeDP-Alg and User-level DP drops, approaching the noiseless AWD-LSTM model's upper-bound result.

• **Non-Sensitive Sentences.** Figures 8 and 10b show that considering non-sensitive sentences (i.e., $q_{\bar{s}} > 0$) significantly helps to improve model utility (i.e., perplexity or test error rate) compared with only considering sensitive-sentences (i.e., $q_{\bar{s}} = 0$). However, different tasks on different datasets may have different optimal values of $q_{\bar{s}}$.



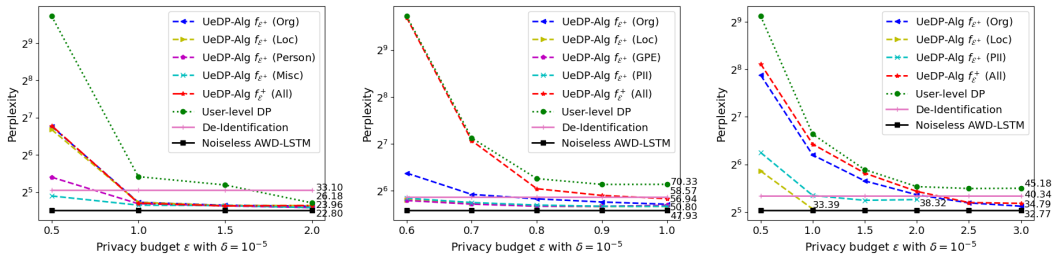
(a) CONLL-2003 dataset (b) AG dataset (c) SEC dataset

Figure 7: Next word prediction results using the GPT-2 model with varying non-sensitive sentences sampling rate $q_{\bar{s}}$ in training. (The lower the better)



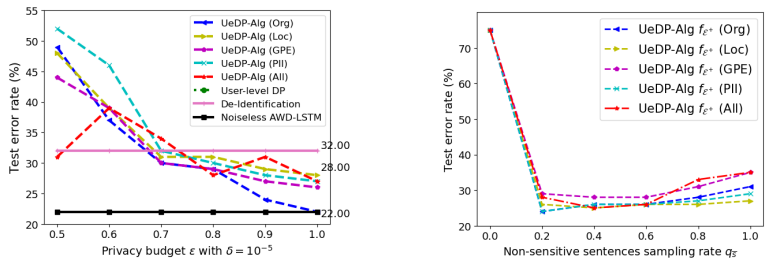
(a) CONLL-2003 (b) AG (c) SEC

Figure 8: Next word prediction results using the AWD-LSTM model with varying non-sensitive sentences sampling rate $q_{\bar{s}}$ in training. (The lower the better)



(a) CONLL-2003 dataset (b) AG dataset (c) SEC dataset

Figure 9: Next word prediction results using the AWD-LSTM model. (The lower the better)



(a) AG dataset (b) AG dataset with varying $q_{\bar{s}}$

Figure 10: Text classification results using the AWD-LSTM model. (The lower the better)